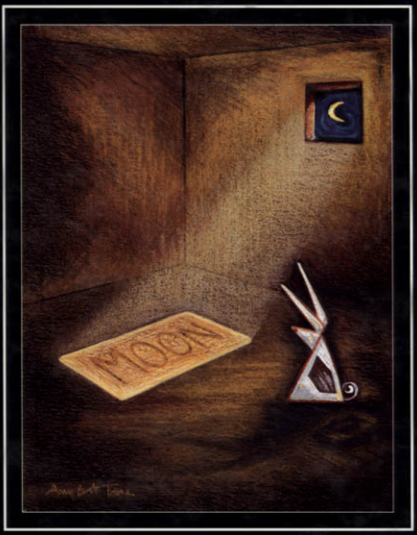
DATA MODEL PATTERNS

Conventions of Thought



DAVID C. HAY

foreword by Richard Barker

DATA MODEL PATTERNS

Conventions of Thought

Also Available from DORSET HOUSE PUBLISHING CO.

Complete Systems Analysis: The Workbook, the Textbook, the Answers
by James and Suzanne Robertson foreword by Tom DeMarco
ISBN: 0-932633-25-0 Copyright ©1994 592 pages, 2 volumes, hardcover

Creating a Software Engineering Culture
by Karl E. Wiegers ISBN: 0-932633-33-1 Copyright ©1996 384 pages, hardcover

Designing Quality Databases with IDEF1X Information Models
by Thomas A. Bruce foreword by John A. Zachman
ISBN: 0-932633-18-8 Copyright ©1992 584 pages, hardcover

Exploring Requirements: Quality Before Design
by Donald C. Gause and Gerald M. Weinberg
ISBN: 0-932633-13-7 Copyright ©1989 320 pages, hardcover

Peopleware: Productive Projects and Teams by Tom DeMarco and Timothy Lister
ISBN: 0-932633-05-6 Copyright ©1987 200 pages, softcover

The Practical Guide to Business Process Reengineering Using IDEF0
by Clarence G. Feldmann foreword by John V. Tieso
ISBN: 0-932633-37-4 Copyright ©1998 est. 264 pages, softcover

Quality Software Management Series by Gerald M. Weinberg

Vol. 1: Systems Thinking
ISBN: 0-932633-22-6 Copyright ©1992 336 pages, hardcover

Vol. 2: First-Order Measurement
ISBN: 0-932633-24-2 Copyright ©1993 360 pages, hardcover

Vol. 3: Congruent Action
ISBN: 0-932633-28-5 Copyright ©1994 328 pages, hardcover

Vol. 4: Anticipating Change
ISBN: 0-932633-32-3 Copyright ©1997 504 pages, hardcover

Surviving the Top Ten Challenges of Software Testing: A People-Oriented Approach
by William E. Perry and Randall W. Rice
ISBN: 0-932633-38-2 Copyright ©1997 216 pages, softcover

What Every Programmer Should Know About Object-Oriented Design
by Meilir Page-Jones foreword by Larry L. Constantine
ISBN: 0-932633-31-5 Copyright ©1995 392 pages, hardcover

Find Out More about These and Other DH Books:

Contact us to request a Book & Video Catalog and a free issue of *The Dorset House Quarterly,* or to confirm price and shipping information.

DORSET HOUSE PUBLISHING CO., INC.
353 West 12th Street New York, NY 10014 USA
1-800-DH-BOOKS (1-800-342-6657) 212-620-4053 fax: 212-727-1044
dhpubco@aol.com http://www.dorsethouse.com

DATA MODEL PATTERNS

Conventions of Thought

DAVID C. HAY

Foreword by Richard Barker

Dorset House Publishing
353 West 12th Street
New York, New York 10014

Library of Congress Cataloging-in-Publication Data

Hay, David C., 1947-
Data model patterns : conventions of thought / David C. Hay.
p. cm.
Includes bibliographical references and index.
ISBN 0-932633-29-3
1. Database design. 2. Data structures (Computer science)
I. Title.
QA76.9.D26H39 1995
658.4'038'011--dc20 95-24983
CIP

Passages reprinted in Chapter 7 from *Accounting* appear courtesy of Richard D. Irwin, Inc. Reprinted by permission. Portions of some models have been previously presented in *Oracle User RESOURCE*, published by the East Coast Oracle Users' Group.

Cover Illustration: Amy McTear
Cover Author Photograph: Richard Chin
Cover Design: Jeff Faville, Faville Design

Distributed in the English language in Singapore, the Philippines, and Southeast Asia by Toppan Co., Ltd., Singapore; in the English language in India, Bangladesh, Sri Lanka, Nepal, and Mauritius by Prism Books Pvt., Ltd., Bangalore, India; and in the English language in Japan by Toppan Co., Ltd., Tokyo, Japan.

Printed in the United States of America

Library of Congress Catalog Number: 95-24983

ISBN: 0-932633-29-3 12 11 10 9 8 7 6 5 4 3 2

Digital release by Pearson Education, Inc., June, 2013

DEDICATION

To Perry Carmichael,
my high school debate coach,
for teaching me how to ask questions
and how to make sense of the answers

This page intentionally left blank

ACKNOWLEDGMENTS

If a man is known by the company he keeps, your author has certainly achieved great fortune. The long list of people I must thank for the ability to write this book begins with my wife, Jola, and my children, Pamela and Bob. (Bob thought the title should be *Fun with Pictures.*) They put up with a lot, and I love them dearly.

The book would not have been possible without Richard Barker, who pioneered the philosophy toward data modeling reflected here. He and my good friend Mike Lynott spent many hours teaching me the approach, and together they inspired me to put it to use enthusiastically. Mike also gets my special appreciation for his encouragement and help with this book.

It has been my privilege to have others to stimulate my imagination and my thinking, and with whom to discuss ideas. This list begins with Ulka Rodgers, who convinced me that I could, indeed, write a book, and who has been of great help with some of the book's thornier chapters. I also thank Chris Bird, Roger Gough, Dave Guthrie, Cliff Longman, Dale Lowery, and Eric Rosenfeld for their many ideas and thought-provoking conversations over the years.

The models presented here are not mine alone. Mike, Cliff, Ulka, and Eric, along with many others, contributed many of the ideas expressed here. Eric was particularly helpful in showing me how to add the quality movement to the contracts, laboratory, and process manufacturing chapters.

My thanks also go to the many people who reviewed parts of the book, including Kathi Bean, Howard Benbrook, John Butler, Howard Eisenstein, Mike Frankel, Mark Gokman, Terry Halpin, John King, Chris Lowde, Sue Peterson, and Becky Winant. And I particularly appreciate Wendy Eakin and David McClintock of Dorset House for their patience and help as I worked my way through this project.

I must of course thank all my employers and clients, each of whom has had something important to teach me in the course of my career. In particular, I would like to single out the Associated Press, which gave me my first opportunity to do a strategic data model for a large, heterogeneous organization. I also thank Texaco, which gave me the biggest organization I've modeled, and the U.S. Forest Service, which provided me with insights about land management and the workings of large Federal agencies.

Tony Ziemba and Adirondak Systems helped me greatly by giving me the *Oracle User RESOURCE* as a forum for the series of articles that provided the seed for this book.

And of course, I thank Oracle Corporation, both for sponsoring the CASE*Method and for giving me the opportunity to learn and promulgate it.

Thanks as well to Continental Airlines, which provided me with "office space" during my weekly three-hour jaunts from Houston to New York and back. Let's hear it for laptop computers and portable disc players!

Finally, I would like to thank the man who, one morning in the summer of 1969, had just cracked up his expensive car in the streets of New York City and was drowning his sorrow in coffee and bagels, when he struck up a conversation with an equally unhappy young man.

As that young man, I was just out of college with my newly minted degree in philosophy. I had come to the Big Apple to seek fame and fortune, but after two very unhappy days, I hadn't found either. Indeed, even the prospect of getting a job was looking pretty dodgy.

After a short conversation, however, the man with the broken car took a chance on me and signed me up to sell a strange new computer service called *time-sharing*. Imagine! As he described it, a customer could attach a teletypewriter to the telephone and call up a computer. Type in a message and it types something back! In 1969, this was clearly black magic, but who was I to argue?

Homer Cates, you got me started in the computer business—way out in left field. It has worked out well for me, and for this I will always be grateful.

CONTENTS

FIGURES AND TABLES

FOREWORD

Entity modelling, or data modelling as it is sometimes called, may be used as a passive way of modelling exactly what exists—providing little interpretation or insight as to its meaning. There is a more active form of modelling, however, commonly found in mathematics and science, which has a model predict something that was not previously known or provide for some circumstance that does not yet exist. Such models are invariably much simpler, easier to understand, and yet deal with more situations than mirror-image models.

For example, the Ptolemy model of planetary motion was complex but accurately described the observable motion of the planets, the moon, and the sun around the earth. The model from Copernicus was simpler but even more accurate, however, giving us the notion of a solar system with planets in motion around the sun. This idea later helped astronomers predict the existence of the previously undiscovered planets Neptune and, years later, Pluto.

If we can model in this sense, using simpler and more generic models, we will find they stand the test of time better, are cheaper to implement and maintain, and often cater to changes in the business not known about initially. For example, rather than just model the exact organisation structure in our business, we could use a generic organisation model that can accommodate executive change (often just whim!). The generic model should even be able to handle the acquisition of another company or a merger with another department—without changing the implementation design. It should be possible simply to declare the new structure to the system.

If used effectively, entity modelling enables a good analyst to talk to users and systems people in their own language and about the issues with which they are concerned. On one hand, a model can be used precisely to articulate the information needs of a business. Used correctly, in discussions with executives and other users, such a model can be used to tease out exceptions that must be dealt with and can then be used to quickly correct misunderstandings, without the analyst having to lapse into technical jargon. On the other hand, the same model can be used in discussions with systems designers to provide them with rich and rigorous definitions of the data. Such models can show much of the processing logic that is implied, not described, by the users. These definitions may be mapped onto relational database designs, with stored procedures and other techniques enforcing the implied processing logic—such as advanced referential integrity constraints. The generic patterns may also be

mapped onto object-oriented designs. This flexibility makes the technique very useful when applied by enlightened practitioners.

Part of the benefit of the approach comes from the layout or positional notation. The notation, described in this book and in my *Entity Relationship Modelling* book, was derived many years ago by Harry Ellis and myself when working on a particularly complex project. We were striving for even greater accuracy in systems analysis, whilst minimising redundant interactions with the users. How could we converge even faster on the desired level of completeness, quality, and simplicity? As a lateral thought, we tried drawing the diagrams in different ways and eventually found the one described herein—often called the "dead crow" notation!

An interesting side effect was that where entities tended to group themselves together on the picture, we often found that they had identical or very similar attributes and relationships. This raised the question, Are they really the same object with different names? This focused question enables the modeller to create a simpler model that caters to *all* the previous concepts already discovered and suggests new ideas that had not been thought of. (It really surprises users when you ask them about things they had forgotten to tell you. You can get responses like, "How did *you* know about that? We only started to think about that last week!") Later, the resulting implementation is usually much quicker and cheaper.

Finally, as you read this book, you may realise that the term "analyst" becomes less and less relevant. It is the concept of "synthesis" that provides the greatest added value to your business—that is, the creation of a model and subsequent system that actively delivers what the business needs for its future success.

In *Data Model Patterns,* David Hay has pulled together many such useful models from his experience and that of the friends and experts that he mentions. If analysts use the well-proven modelling approach described in this book, and then implement the results on relational or object database management systems, they should be able to develop highly business-oriented systems quickly.

May 1995
Near-Maidenhead
Berkshire
England

Richard Barker
Senior Vice President
Product Division
OpenVision
Pleasanton, California

PREFACE

Learning the basics of a modeling technique is not the same as learning how to use and apply it. Data modeling is particularly complex to learn, because it requires the modeler to gain insights into an organization's nature that do not come easily. For example, an analyst may be expected to come into an organization and immediately understand subtleties about its structure that may have evaded people who have worked there for years.

This book is intended to help those analysts who have learned the basics of data modeling, but who are looking for help in discovering subtleties and in obtaining the insights required to prepare a good model of a real business. Moreover, the book is intended to help analysts produce models that are easier to read, by virtue of standards of diagram structure and organization.

The book is based on the assumption that the underlying structures of enterprises are similar, or at least that they have similar components. Understanding those similarities gives an analyst a starting model, which can then be massaged and adjusted as necessary to match the specific circumstances of a particular company. This is not to say that all companies' models will look the same. Quite the opposite is true. In your author's experience, no two organizations' models have been identical. On the other hand, widely differing organizations, from government health protection agencies to oil refineries, have many similar components.

An analyst who has these components in his intellectual tool kit is in a good position to grasp quickly what is unique about an enterprise and to draw a data model that both embodies universal truths and specifically represents the business at hand.

This book has a second audience as well: As a child of the Sixties, I got into the business world only reluctantly. Among the problems I faced was understanding just how business works. Even in business school, I was never able to find an introductory-level course that described how it works as a whole. Each course analyzed a specific area in detail, but none really provided the overview I sought. It was only as I saw the patterns expressed in the structure of business information that a business uses, that I began to come to grips with the issues involved. Perhaps this book can be useful to a similarly disadvantaged student trying to understand the nature of the business world.

April 1995
Houston, Texas

D.C.H.
Davehay@essentialstrategies.com

This page intentionally left blank

3

THE ENTERPRISE AND ITS WORLD

We'll start the modeling exercise with some entities that can be specified without interviewing anyone. You know that they will be required, no matter what the business.

An enterprise cannot exist without people. Whether an employee, a vendor agent, or the president of a company, a PERSON can be assumed to be a "thing of significance" to most companies. It should be no surprise, therefore, that the PERSON entity will appear on virtually all data models for all companies. Oh, there may be pressure to name it something different, like EMPLOYEE, CUSTOMER, AGENT, or whatever. But ultimately, the thing of significance is a PERSON, and as a modeler, you will save a lot of time by simply putting that entity on the diagram before you start interviewing.

A few of the things to be known about a person, such as "name" or "birth date," are attributes of the PERSON entity. Others (more than you might suppose) are not actually attributes, but are relationships to other entities, as we shall see. A person may enroll in one or more courses, for example, or may play a role in one or more activities. (More examples of these roles will be shown throughout the book.)

If an enterprise is concerned with people, it must surely also be concerned with aggregations of people. An ORGANIZATION, then, must also be a thing of significance to nearly any enterprise. An ORGANIZATION may be a department, a committee, a vendor, a labor union, or any other collection of people or other organizations. It is described by such attributes as "purpose," "Federal tax ID," and so forth.

Again, save yourself some time. Put ORGANIZATION on your model even before you start to interview.

PARTIES

People and organizations share many attributes and many relationships to other entities. A corporation is, after all, a "legal person." Both people and organizations have "names" and "addresses" as attributes, and both may be parties to contracts. For this reason, while PERSON and ORGANIZATION are things of significance, so too is the super-set of the two, which we shall here call PARTY. This is shown in Figure 3.1.

So, we now have three entities—among the most important entities on our model—and we haven't spoken to anyone yet!

To get attributes, however, we will have to interview someone. Enterprises differ greatly in the kinds of information they want to hold about people and organizations. Even here, though, we can make some educated guesses. For example, PARTY probably has the common attributes "name" and "address," while PERSON has the attribute "birth date," and ORGANIZATION may have an attribute such as "purpose." Now while both PERSON and ORGANIZATION have "names," however, PERSON actually has two names (plus a middle name or initial, if you want to get thorough*). This could be handled by moving "name" to ORGANIZATION and giving PERSON "first name" and "last name." An alternative is shown here, with the principal "name" being equivalent to a person's surname, and with only the given name specific to PERSON. How this should be handled in your model depends on the requirements of the organization.

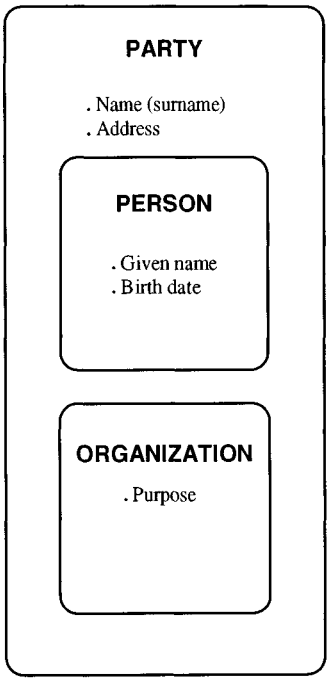


Figure 3.1: Parties.

* In the United States, that is . . . most of the time . . . but not counting George Herbert Walker Bush. Many other cultures use multiple middle names, plus "de," "von," "van," "la," and the like. If the model describes an organization operating entirely within the United States, assumptions can be made about names. If the organization is multinational, these attributes have to be made more general.

One premise that must be established when generalizing models is the context in which this is to happen.

While virtually all companies and government agencies have need for the PERSON and ORGANIZATION entities, how these entities are divided into subtypes will vary from company to company. (At this point, you really do have to speak to someone.)

Beginning with the PERSON entity, a common practice is to assert that a PERSON may be either an EMPLOYEE or an OTHER PERSON. (See Figure 3.2.) An EMPLOYEE is usually thought of as a thing of significance to an employer.

From a practical point of view, this can work, especially since employees often have an extensive set of attributes that don't apply to other people. It turns out, for example, that "birth date" is probably an attribute of interest only for an employee, and there are others, such as "social security number," "number of exemptions," and so forth.*

There are problems with defining EMPLOYEE as an entity, however: First, a person may fall into more than one of these categories. That person may have worked for a customer as an agent and is now an employee—or vice versa—but the "agentness" of the person is to be kept. Consultants and other contractual workers are also problematic, since a lot of employee-type information may be held, even though such people are not, strictly speaking, employees. If these are significant issues in your organization, then the EMPLOYEE/OTHER PERSON distinction is *not* appropriate.

This is one example, by the way, of a common trap for the unwary. (We will encounter others.) What you have in the word "employee" is a common name for something including in its meaning not just the thing itself, but also its *relationship* to something else. A PERSON is a human being with specific characteristics, whether employed by anyone or not. An EMPLOYEE, on the other hand, is a PERSON who has established a relationship of employment with an ORGANIZATION. Figure 3.3 shows this relationship: Each PERSON may be *currently employed by* one and only one ORGANIZATION, and each ORGANIZATION may be *the employer of* one or more PEOPLE.

* Note to readers outside the United States: Federal income taxes are adjusted according to the number of people in the family. An "exemption" from a certain amount of tax is granted to each member, and additional exemptions are granted to people in special circumstances. The number of exemptions an employee declares affects the amount of money withheld from each paycheck for taxes.

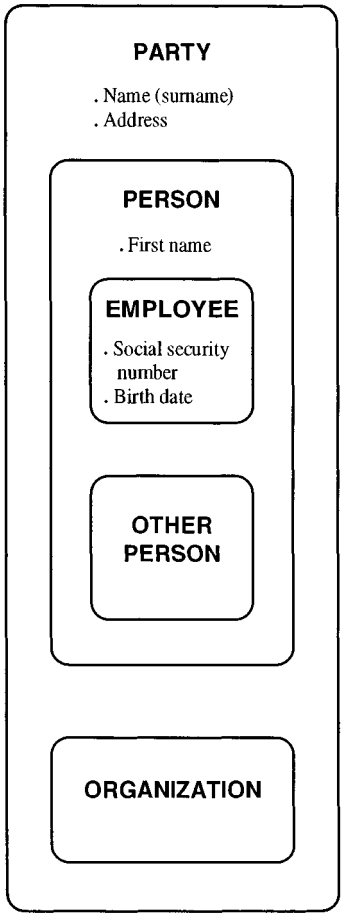


Figure 3.2: Employees.

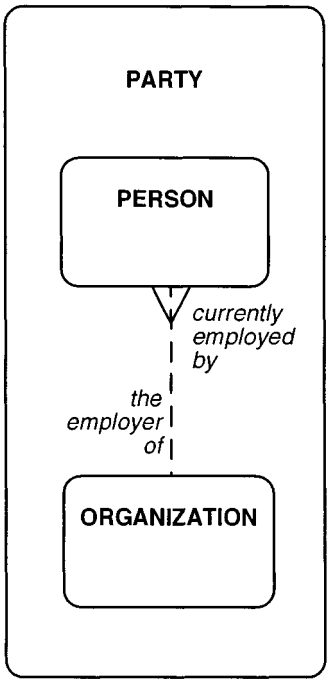


Figure 3.3: The Employment Relationship.

Showing it this way, however, raises the question of what to do with the employment-specific attributes. It is also probably true that over time, a PERSON may be employed by more than one ORGANIZATION. For these reasons, an additional entity probably will be required to describe the relationship fully. Figure 3.4 shows the addition of the entity EMPLOYMENT, to solve both problems. Note that each EMPLOYMENT must be *of* a PERSON *with* an ORGANIZATION. Each PERSON may be *in* one or more EMPLOYMENTS, and each ORGANIZATION may be *the source of* one or more EMPLOYMENTS.

Employment attributes such as "number of exemptions" go in this entity. EMPLOYMENT may also have the attribute "type," to distinguish between full-time employees, part-time employees, and contractors. It may be argued that "birth date" or "social security number" should go there, since they are of

interest only in the context of employment. In fact, however, these are attributes of PERSON, and should be placed in that entity. If your client can claim *never* to want to know the birth date of a nonemployee, you might get away with making them attributes of EMPLOYMENT—but when the time comes that you *do* want to know the birthday of a nonemployee ("I know what I said, but that was then . . ."), you will appreciate that you did the more philosophically correct thing and made it an attribute of PERSON.*

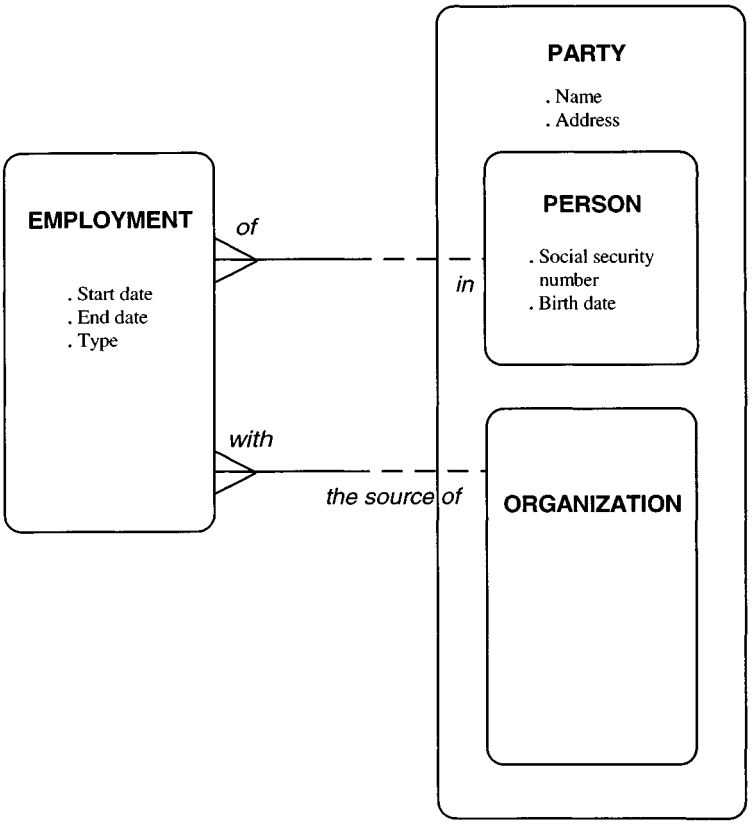


Figure 3.4: The Employment Entity.

* Of course, by making "social security number" an attribute of PERSON, you promote the insidiously infectious practice common in the United States of collecting social security numbers for everyone, in all kinds of inappropriate situations. Here, political views on privacy may conflict with those of a modeling purist.

Note that there is some ambiguity in this model. The model does not and cannot make explicit an assumption probably made by viewers—that a PERSON will be *in* only one EMPLOYMENT at a time. One could assume that the "more than one" EMPLOYMENTS are a succession of jobs *of* the PERSON. The model does not prevent, however, the PERSON from having multiple EMPLOYMENTS at once, possibly even with different ORGANIZATIONS. Constraints to prevent such a situation are business rules, which would have to be specified outside the model.

EMPLOYEE ASSIGNMENTS

Figure 3.4 shows that each PERSON may be *in* one or more EMPLOYMENTS *with* an ORGANIZATION. A PERSON's complete relationship to an ORGANIZATION, however, can be more complex. Figure 3.5 shows this.

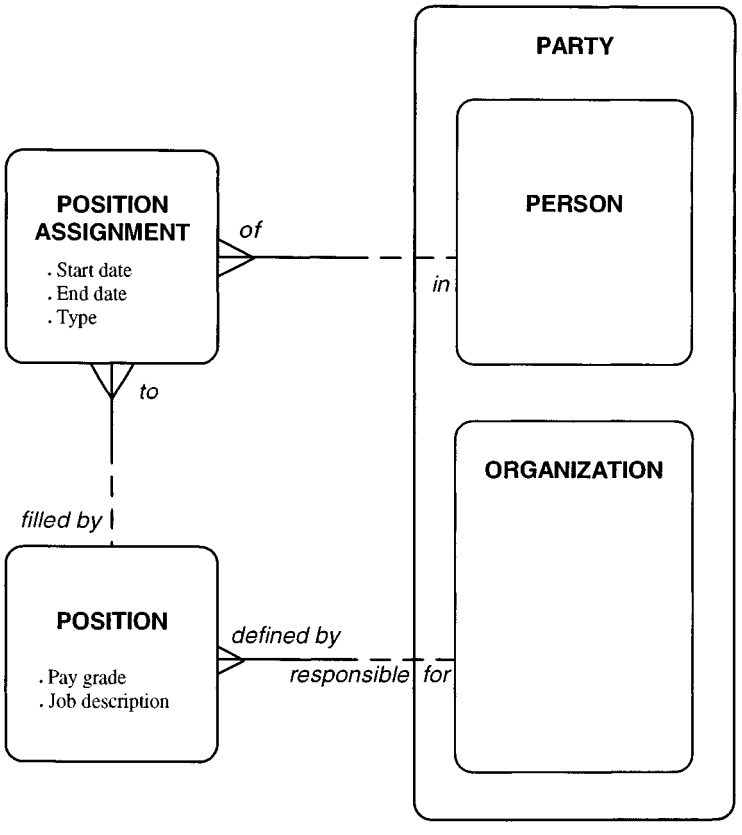


Figure 3.5: Positions.

First, the POSITION held by a person is itself something of significance, with attributes such as "pay grade" and "job description." Typically *defined by* one

ORGANIZATION, the POSITION is likely to be held by more than one PERSON, at least over time, and a PERSON may also reasonably be expected to hold more than one POSITION over time. Indeed, the PERSON may hold multiple POSITIONS at the same time. For example, a scholar might progress through the titles of Teaching Assistant, Assistant Professor, and Professor. While holding the last title, the Professor might become a Department Chairman as well.

All this argues for specification of POSITION ASSIGNMENT (the fact that a PERSON holds the POSITION, for a period of time, beginning with the "start date" and lasting until the "end date"). That is, each POSITION ASSIGNMENT must be *of* a PERSON *to* a POSITION. Each PERSON, then, may be *in* one or more POSITION ASSIGNMENTS, each of which must be *to* a POSITION *defined by* an ORGANIZATION.

There are many variations on this model. If, for example, POSITIONS are defined company-wide, and departments use them with different TITLES, the situation is as shown in Figure 3.6. Each POSITION ASSIGNMENT must be *to* a TITLE, not a POSITION. This TITLE, in turn, must be *for* a POSITION, and the TITLE, not the POSITION, is *defined by* an ORGANIZATION, which in this context is a department.

Most uses of TITLE and POSITION might be expected to be concerned only with INTERNAL ORGANIZATIONS. The relationship is shown connected to ORGANIZATION, however, because it may be important to keep track of titles and organizational structures in other companies as well.

The POSITION ASSIGNMENT, TITLE, and POSITION entities need not be limited to formal employment. In companies that use "matrix management" techniques, where a person plays a role for many different departments, a PERSON may have a permanent assignment to one department while being *seconded* to another.* A second POSITION ASSIGNMENT (of "type" "secondment") for the same person would then be specified.

* "Seconded" is a British term for "temporarily assigned."

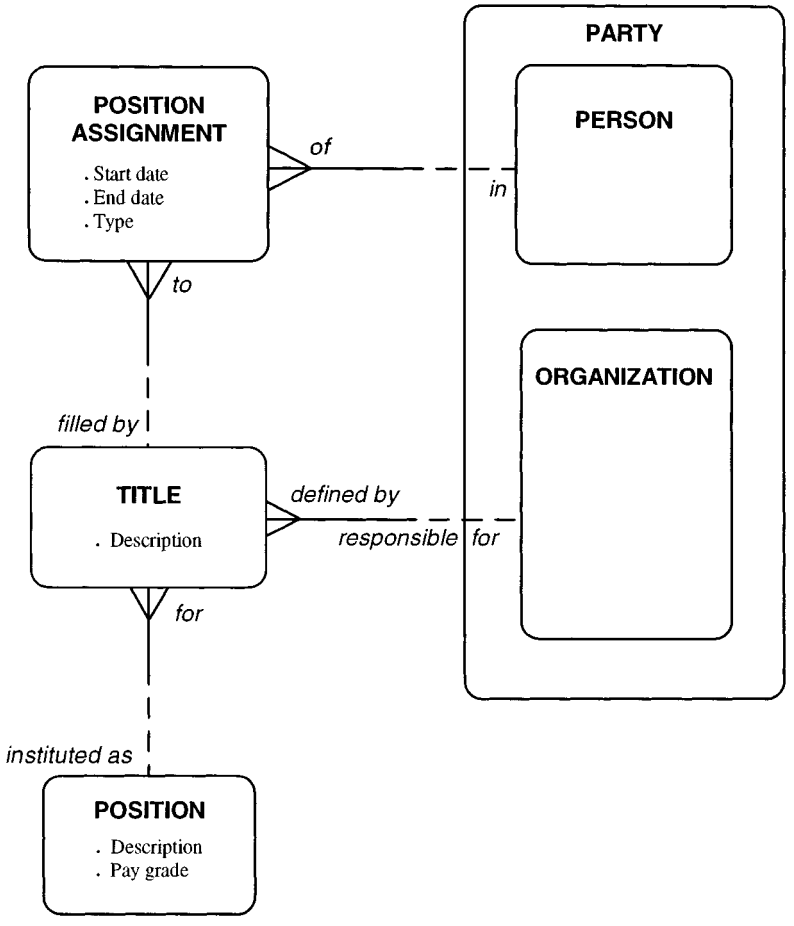


Figure 3.6: Titles.

If your organizational situation calls for a more elaborate model, PARTY and POSITION ASSIGNMENT entities could be related to the EMPLOYMENT entity introduced in Figure 3.4, thereby producing the diagram shown in Figure 3.7. POSITION ASSIGNMENT is now *based on* the EMPLOYMENT *of* the PERSON *with* an ORGANIZATION. Note that this new model allows EMPLOYMENT to be *with* one organization (such as a company), while a POSITION may be *defined by* a different ORGANIZATION (such as a department). This representation would allow a PERSON to have EMPLOYMENT *with* one company and to have a POSITION *defined by* an unrelated ORGANIZATION. This includes, for example, the seconding example cited

above, exhibited in government agencies where employment might be defined for one agency but the person is temporarily assigned to another.

This also describes a consultant who is employed by a consulting company but assigned to a POSITION in a client company.

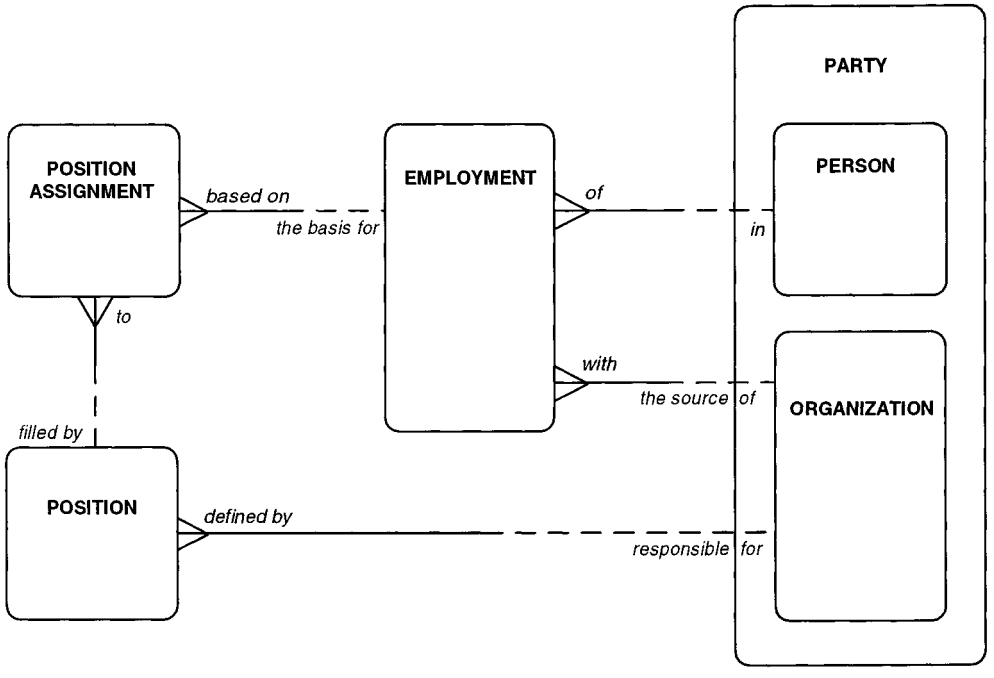


Figure 3.7: Employment (Revisited).

ORGANIZATIONS

As with PERSON, the specific nature of the organization being modeled dictates the way the entity ORGANIZATION is divided into subtypes. To resolve this, again, we actually have to interview someone.

A common approach to ORGANIZATION is to divide it into INTERNAL ORGANIZATION and EXTERNAL ORGANIZATION.

An INTERNAL ORGANIZATION could be, for example,

- a DEPARTMENT,
- a DIVISION, or
- some OTHER INTERNAL ORGANIZATION.

Figure 3.8 shows this, as well as the world of the EXTERNAL ORGANIZATION, which could be

- a CORPORATION,
- a GOVERNMENT AGENCY, or
- some OTHER EXTERNAL ORGANIZATION.

Again, these are just examples, and your model may be different. While it is not shown here, these subtypes could be broken down further. GOVERNMENT AGENCY, for example, could be divided into

- FEDERAL GOVERNMENT AGENCY,
- STATE GOVERNMENT AGENCY,
- LOCAL GOVERNMENT AGENCY, and
- FOREIGN GOVERNMENT AGENCY.

This kind of detail is only necessary in certain situations, however. Professional associations, labor unions, and other agencies may also be added.

Note once again that, when we split out kinds of organizations, attributes of EXTERNAL ORGANIZATION, such as "purpose," apply to all external organizations, but attributes of each subtype (such as "Federal tax ID" in CORPORATION) apply only to that subtype.

For purposes of this example, "number of employees" is shown as an attribute of INTERNAL ORGANIZATION. It is a particularly weak example, however, and you may be able to think of a better attribute that is specific to INTERNAL ORGANIZATION. It may, for example, be the case that you want to record the "number of employees" for EXTERNAL ORGANIZATIONS, as well. In fact, for INTERNAL ORGANIZATIONS, this may be a derived attribute, obtained for an occurrence by simply counting the number of occurrences of EMPLOYMENT that are *with* that occurrence.

Note also that this view of INTERNAL and EXTERNAL ORGANIZATIONS, like the inclination to define EMPLOYEE as a subtype of PERSON, implies a strong orientation toward a world divided between "us" and "them." A subtype FOREIGN GOVERNMENT AGENCY, for example, suggests that all governments except the one in our home country is foreign. This is the view taken by many companies, but the modeler should be aware of its bias. For an international company, governments are governments and each division may have its own definition of what is "foreign." This would make FOREIGN GOVERNMENT AGENCY a meaningless entity.

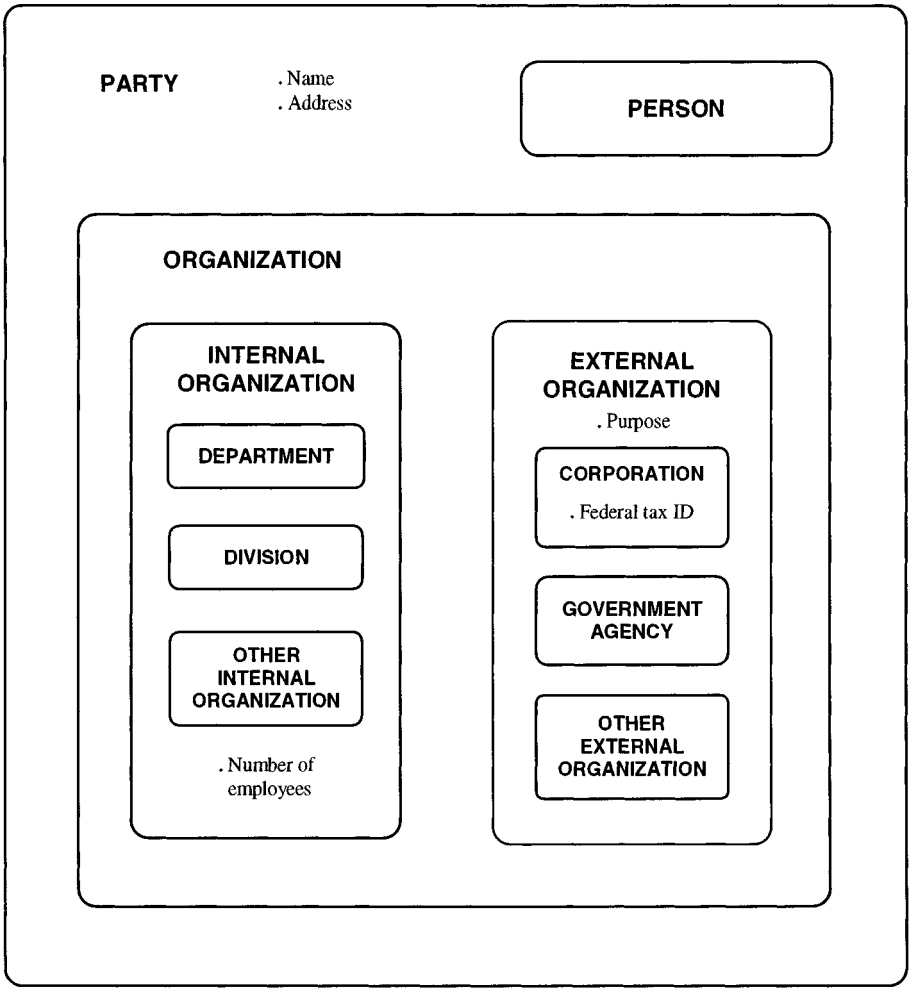


Figure 3.8: Organizations.

ADDRESSES

PARTIES are usually located somewhere. In its simplest form, the model could simply include "address" as an attribute of PARTY, as shown in Figure 3.1. (This would be inherited by both PERSON and ORGANIZATION.) The problem with this is that organizations at least, and many people too, have more than one address, such as "shipping address," "billing address," "home address," and so forth.

This argues for adding a second entity ADDRESS, as shown in Figure 3.9. Here, each ADDRESS must be *the location of* one and only one PARTY. Each PARTY, in turn, may be *at* one or more ADDRESSES. Attributes for ADDRESS include the "text" of the address, plus at least "city," "state," and "postal (ZIP) code."* Alternatively, ADDRESS might have only the attribute "city, state, and postal code" as a single string. In either case, ADDRESS should also include as an attribute address "type," that could be "billing address," "shipping address," "home address," and so forth.

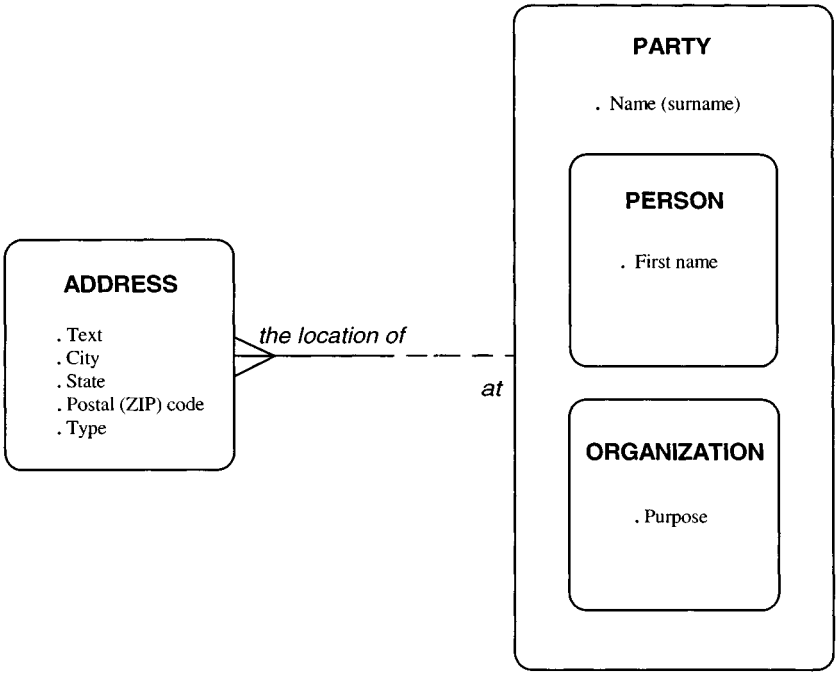


Figure 3.9: Addresses—First Try.

So, we have now made ADDRESS a thing of significance. This is not unreasonable, if you think of an office, a home, or a work center. Making ADDRESS a thing of significance, however, leads to a problem: We have asserted that each ADDRESS must be *the location of* one and only one PARTY, but when you think about it, more than one PERSON or ORGANIZATION can be at the same ADDRESS.

* The context of the model will determine whether this attribute is "ZIP code" or "postal code." If the client organization will operate entirely within the United States for the foreseeable future, the assumption of a nine-digit, two-part numeric "ZIP code" can be made. If not, "ZIP code" must become "postal code" and no formatting assumptions are possible.

Indeed, the word "address" is ambiguous: In ordinary conversation, an address may be associated with a single party ("What's Steve's address?"), but in other contexts, it can also be associated with multiple parties. ("The Grand Junction, Colorado, office at 2476 Galley Lane employs twenty people.") We have not expressed the concept of address as an identified place.

To clarify this, we will rename the ADDRESS entity as SITE. A SITE (such as the Grand Junction office) is a place with a designated purpose. It is not a geographic location (like the city of Grand Junction, itself); that is simply a location on a map. A SITE may be an office, a work center in a factory, a warehouse location, or an archaeological dig. The key word in the definition is "purpose."

In this example, we can distinguish between the attribute "purpose"—that might be "administration," "manufacturing," "storage," and the like—and "type," which describes the kind of site it is, such as "office," "work center," or "warehouse location." This distinction represents a purist approach: In some circumstances, one attribute might cover both the type and the purpose of the SITE.

Because more than one PERSON or ORGANIZATION may be located *at* a SITE, we now need an entity to represent each fact that a PARTY is located *at* a SITE. We could call this ADDRESS, and in many companies' models, it is. Because of the ambiguity described above in the way we use the word "address," however, perhaps it would be better to invent a new word. In Figure 3.10, it is shown as PLACEMENT. Each PLACEMENT must be *of* a PARTY *at* a SITE.

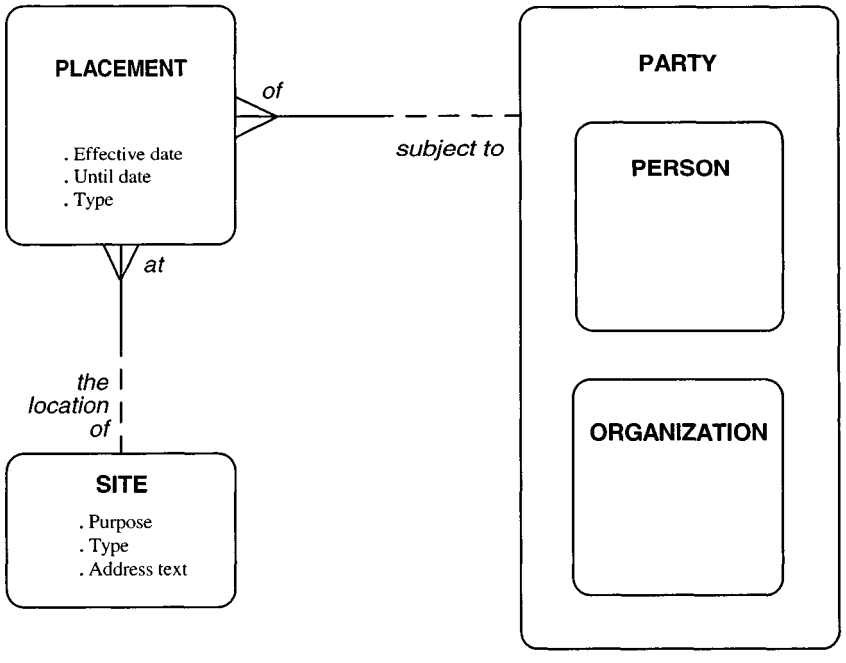


Figure 3.10: Site (Another Way to Show Addresses).

Attributes of PLACEMENT include the date it happens ("effective date") and the date it is discontinued ("until date"). PLACEMENTS may also be categorized via the attribute "type."

To summarize, then, each PARTY may be *subject to* one or more PLACEMENTS *at* a SITE. That SITE determines the PARTY'S "address." That is, a PARTY will have one "address" for each SITE where it is located.

GEOGRAPHIC LOCATIONS

Such a solution may be adequate for many applications, but often it is of interest to collect addresses by city, county, postal code, or other GEOGRAPHIC LOCATION where the address is located. Each SITE, then, must be *in* one GEOGRAPHIC LOCATION—which means that each GEOGRAPHIC LOCATION may be *the location of* one or more SITES. (See Figure 3.11.)

The attributes of GEOGRAPHIC LOCATION would of course include its "name," and possibly a "geographic location type," such as "state," "country," "province," and so forth.

With all this, though, what about our original problem of modeling a mailing address? Unfortunately, the answer is, "It depends."

If GEOGRAPHIC LOCATION simply located the SITE in general terms, the "address text" and "city, state, and postal code" attributes could remain in SITE. GEOGRAPHIC LOCATION is itself hierarchical, however, where each GEOGRAPHIC LOCATION may be *part of* one and only one other GEOGRAPHIC LOCATION.* Thus, you may include all the countries, states, provinces, cities, postal codes, neighborhoods, major statistical metropolitan areas (MSMA's), and streets as examples of GEOGRAPHIC LOCATION and then link them together as a hierarchy. Correctly populating the GEOGRAPHIC LOCATION entity would then make redundant at least the "city, state, and ZIP code" in SITE. A building at "544 East 11th Street, New York, New York 10009," for example, could have as the "address text" of its SITE "544 East 11th Street," and then be shown as being *in* ZIP code "10009," which is *part of* "New York" (the city), which is *part of* "New York" (the state).

Indeed, carried to extremes, even "East 11th Street" could be a GEOGRAPHIC LOCATION, as could "building 544," which is *part of* "East 11th Street." The only "address text" required in SITE, then, would be the apartment number that uniquely identifies the PARTY'S location. (In this case, the PARTY is probably an ORGANIZATION of "type" "family.")

The purest answer, then, in terms of the logic of the model, would be to put most of the address in successive GEOGRAPHIC LOCATIONS as shown, with only the most detailed element defined in SITE.

---

* This kind of relationship is called "recursive." In *Hay's First New Dictionary,* the entry for this word is, "recursive—*see recursive.*"

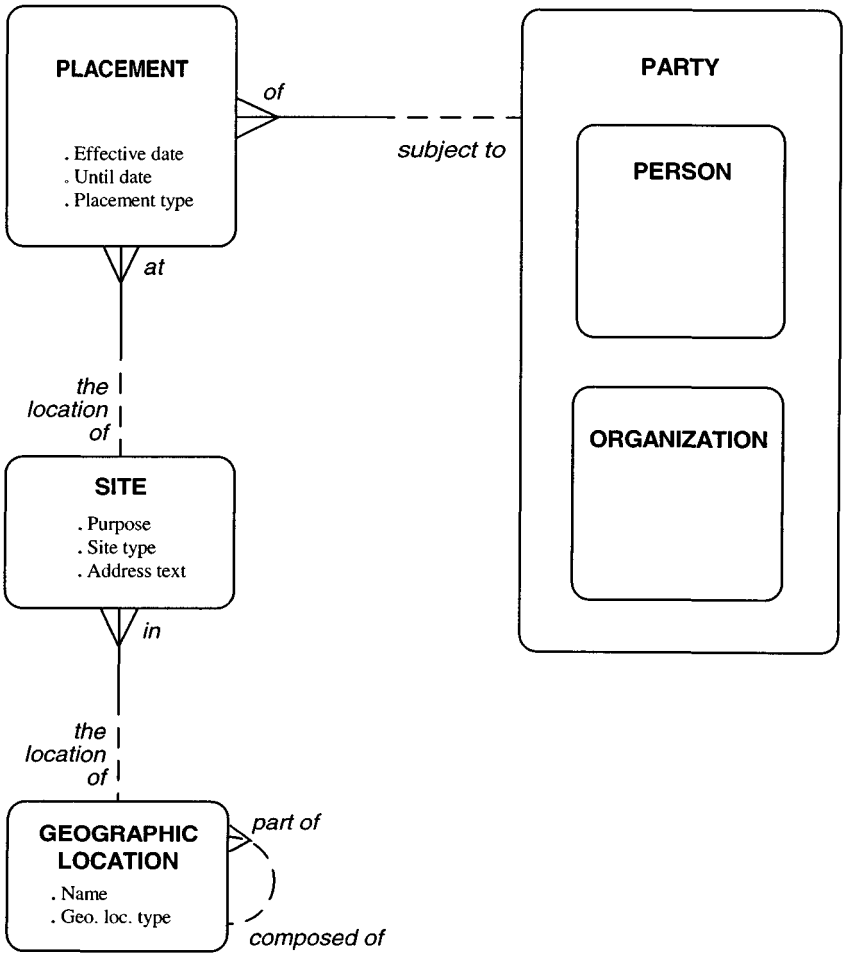


Figure 3.11: Geographic Location.

If all we ever will want are mailing addresses for invoices and labels, however, this latter approach is clearly overdoing it. For mailing labels, an address is a single piece of information. This puts us back to the original model with three text attributes in ADDRESS or PARTY. ADDRESS could still point to the appropriate GEOGRAPHIC LOCATION for classification purposes, although it should be understood how this introduces redundancy in the data. (This is not necessarily bad, as long as the business understands it and takes responsibility for it.)

Note that this deference to practicality is not an example of concern for more efficient computer processing. It is recognition of the fact that the data

mean different things to different people. If, to a user community, "address" is a single attribute, the model should reflect that—as long as that community is made aware of the possibility that this view could change in the future, and the implications of that change. If, on the other hand, the user community is interested in using "address" in many different ways, the more complex model would be more appropriate.

The address data model provides an example of a case in which there is no *right* model. The final product must reflect not only the underlying structure of the data, but also the view of that data held by the organization. Specifically, we must ask the question, "What are the things of *significance* to the company or agency?"

Having asked that, however, we must note that both views may be held in the same organization: One department only wants mailing labels, while another wants to do detailed geographic market research. In such a case, the modeler must go with the more conceptual approach and, when the system is implemented, provide for the more mundane user needs through application "views."* The more complex model is truer to the conceptual structure of the data, and can accommodate all the other perspectives. None of this, by the way, addresses what the final physical database structure will look like. The designer must be the final arbiter of what will actually work in the organization's computers.

GEOGRAPHIC LOCATION can also be exploded into considerably more detail. A large part of the mission of the USDA Forest Service, for example, is to manage land. Thus, it is concerned with all kinds of real estate. The GEOGRAPHIC LOCATION for the Forest Service can be generalized to mean any kind of LAND PARCEL. Figure 3.12 shows this. In this view, a LAND PARCEL includes: GEOPOLITICAL LAND PARCELS, such as the STATES, CITIES, and COUNTIES discussed above; MANAGEMENT AREAS, such as NATIONAL FORESTS, FOREST SERVICE REGIONS, and other ADMINISTRATIVE AREAS; SURVEYED LAND PARCELS, described in terms of TOWNSHIPS, SECTIONS, and so forth; and designated NATURAL AREAS, such as HABITATS and other areas with common natural characteristics. In fact, in the actual Forest Service model, many of the subtypes are broken down even further.

The hierarchical relationship shown in Figure 3.11, that one GEOGRAPHIC LOCATION is *part of* another, has common application throughout the model. We first saw this pig's ear symbol in Chapter Two, and we will encounter it again frequently. Note that the hierarchical relationship must be all optional, since if you said, for example, that "each GEOGRAPHIC LOCATION *must be part of* one and

* That is, program logic could be *implemented* so that a particular user is shown a "view" of a PARTY (or PERSON or ORGANIZATION, as appropriate) table (entity), with the column (attribute) "address," as though it were derived from the models in either Figure 3.5 or Figure 3.1. In fact, however, when asked for "address," the program traverses other tables to retrieve it.

only one other GEOGRAPHIC LOCATION," there would be no way to deal with the GEOGRAPHIC LOCATION at the top. Similarly, to say that each GEOGRAPHIC LOCATION must be *composed of* one or more GEOGRAPHIC LOCATIONS is to fail to deal with the bottom of the tree.

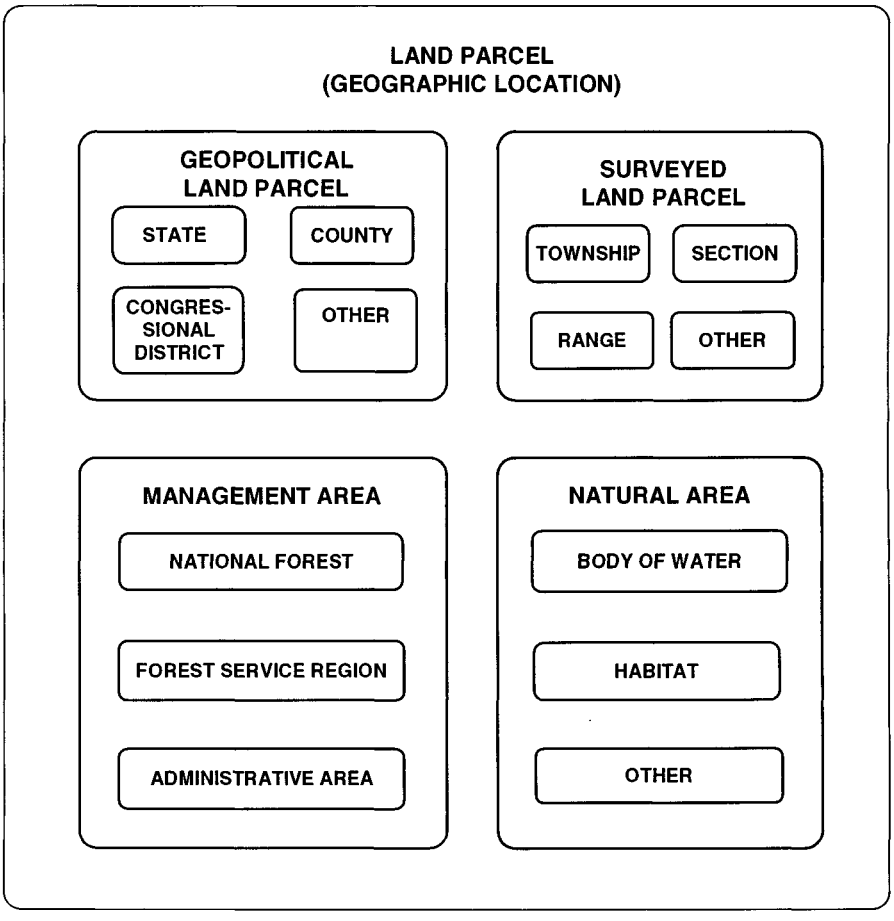


Figure 3.12: Land Parcels.

Note also that this relationship contains a business rule assumption that must be documented somewhere else: There is nothing in the model to prevent a GEOGRAPHIC LOCATION from being specified as part of itself, either directly or through a chain (that is, A is part of B, which is part of C, which is part of A). This is the case when any hierarchy is represented like this.

The idea of GEOGRAPHIC LOCATION itself can get trickier than this, and again, how it is modeled depends entirely on how sophisticated the company wishes to be in dealing with geography.

Life is more complicated, for example, in those cases in which geography is not strictly hierarchical. Cities are usually inside counties, except for New York

City, for example, which has five counties inside it. Also, in the United States, a ZIP code is normally entirely within a city, but not always: In Oregon, ZIP code 97401 encompasses both Coburg and part of Eugene. Eugene also has several other ZIP codes within it.

In another example, a project dealing with Native Canadians required a model to deal with the case in which a tribal land covered portions of more than one province. The land could not be considered inside the province, and the province was certainly not inside the tribal land.

These examples make it necessary to define an additional entity, GEOGRAPHIC STRUCTURE ELEMENT, each occurrence of which would describe the fact that part of one GEOGRAPHIC LOCATION is part of another. This is shown in Figure 3.13. Each GEOGRAPHIC LOCATION may be *composed of* one or more GEOGRAPHIC STRUCTURE ELEMENTS, each of which must be *the presence of* one other GEOGRAPHIC LOCATION. Alternatively, each GEOGRAPHIC LOCATION may be *a part in* one or more GEOGRAPHIC STRUCTURE ELEMENTS, each of which must be *in* another GEOGRAPHIC LOCATION.

In the Native-Canadian land case, for example, each occurrence of a tribal land parcel's existence in a province constitutes one GEOGRAPHIC STRUCTURE ELEMENT. That is, if a tribal land were in Quebec and Ontario, the land's GEOGRAPHIC LOCATION would be *composed of* two GEOGRAPHIC STRUCTURE ELEMENTS—one representing *the presence of* part of the land in Quebec, and one representing *the presence of* part of it in Ontario.

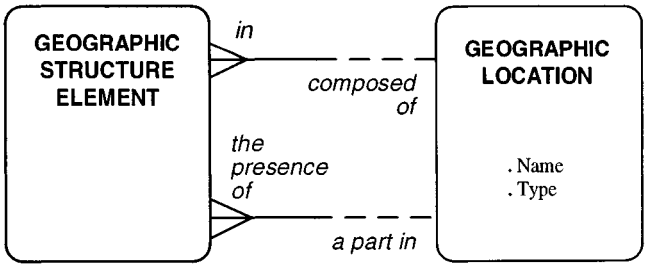


Figure 3.13: Geographic Structure Elements.

REPORTING RELATIONSHIPS

EMPLOYMENT and POSITION ASSIGNMENTS are examples of relationships that may exist between PEOPLE and ORGANIZATIONS. There are in fact many others—too many to be modeled as specifically as this. Consequently, we need a more general approach to relating PEOPLE and ORGANIZATIONS to each other. Such an approach is presented in this section.

Figures 2.5 and 2.6 of Chapter Two showed the derivation of the ORGANIZATION and its hierarchical structure. Part of Figure 2.6 is reproduced here as Fig-

ure 3.14, showing that each ORGANIZATION may be *composed of* one or more other ORGANIZATIONS.

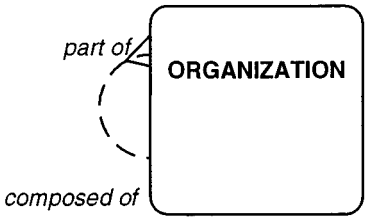


Figure 3.14: Organizational Structure.

In Chapter One, we ended the discussion of ORGANIZATION when we established that all organizations are fundamentally the same. It is important, however, that we also know how to draw a hierarchy when the top (or bottom) element (in this case, ORGANIZATION) is significantly different from the others.

For example, we might wish to assert that a CORPORATION is fundamentally different from an OTHER ORGANIZATION. This distinction may be expressed in either of two ways: The first is shown in Figure 3.15. In this, each OTHER ORGANIZATION must be *part of* one and only one OTHER ORGANIZATION or it must be *part of* a CORPORATION. Note that we can now say each ORGANIZATION must be *part of* another ORGANIZATION, since at the top of the hierarchy the other side of the arc takes effect. For that last step, each ORGANIZATION must be *part of* a CORPORATION. We still must say "may be" going *down* the hierarchy ("each OTHER ORGANIZATION may be *composed of* one or more OTHER ORGANIZATIONS"), since there is no defined bottom to it.

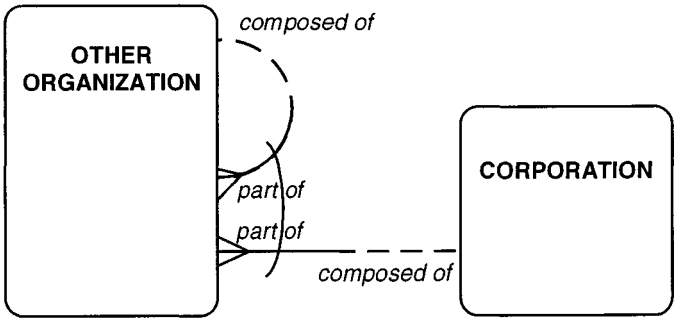


Figure 3.15: Top-Heavy Hierarchy—Version 1.

The second way a distinction may be drawn between the top element in a hierarchy and all the others (which is either more elegant or more arcane, depend-

ing on your taste) is shown in Figure 3.16. In this diagram, each OTHER ORGANIZATION must be *part of* an ORGANIZATION (which in turn must be either a CORPORATION or an OTHER ORGANIZATION). Going the other way, each ORGANIZATION (whether it is a CORPORATION or an OTHER ORGANIZATION) may be *composed of* only one or more OTHER ORGANIZATIONS.

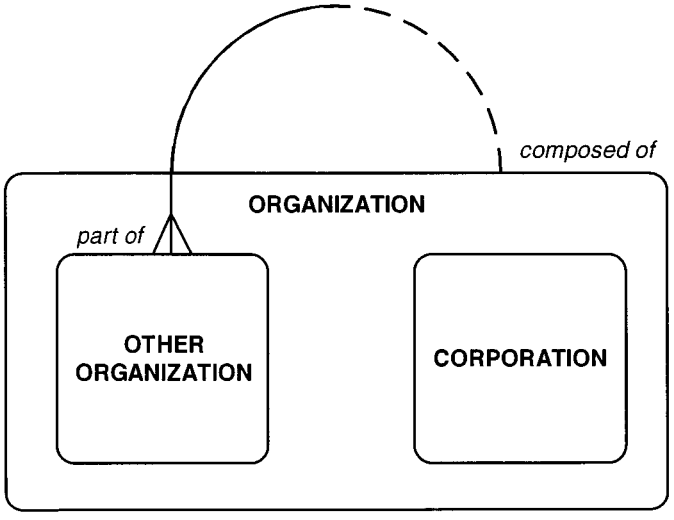


Figure 3.16: Top-Heavy Hierarchy—Version 2.

All of this is well and good, until you start dealing with a government agency that, over time, has been part of several different departments and other agencies. It turns out *not* to be the case that an ORGANIZATION may be *part of* only *one* ORGANIZATION. The pig's ear turns out to represent a many-to-many relationship. This requires us to add an entity describing each occurrence of an ORGANIZATION being part of another. The entity added is another example of a "structure" entity, like the GEOGRAPHIC STRUCTURE ELEMENT discussed above. In this case, we have added REPORTING RELATIONSHIP in Figure 3.17. Each REPORTING RELATIONSHIP must be the occurrence *of* one ORGANIZATION *in* another ORGANIZATION. That is, each ORGANIZATION may be *composed of* one or more REPORTING RELATIONSHIPS, each of which is *of* another ORGANIZATION.*

* As with the hierarchy, we will stipulate outside the model that an ORGANIZATION may not be related to itself. That is, an occurrence of a REPORTING RELATIONSHIP may not be both *in* and *of* the same ORGANIZATION.

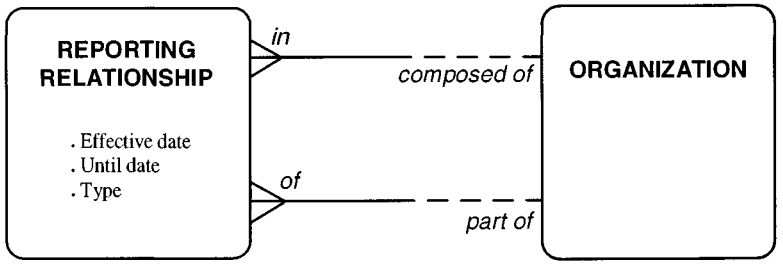


Figure 3.17: Reporting Relationships.

Having established REPORTING RELATIONSHIPS between ORGANIZATIONS, we face another issue. It is possible and often necessary to describe relationships between PEOPLE as well, and relationships between PEOPLE and ORGANIZATIONS.

We've already seen EMPLOYMENT as one relationship between PEOPLE and ORGANIZATIONS. Because this relationship is often referred to directly, and requires special treatment, it was modeled explicitly. This is only one example, however, of a relationship that can exist between two parties. People are married to each other; people belong to unions and clubs; departments are contained in divisions; and companies band together into industrial associations, buying groups, and so forth.

For this reason, we have generalized REPORTING RELATIONSHIP in Figure 3.18 to cover *any* relationship between two PEOPLE or ORGANIZATIONS. We also generalized the relationship names, to say that each REPORTING RELATIONSHIP must be *from* one PARTY *to* another. Conversely, a PARTY may be *on one side of* one or more REPORTING RELATIONSHIPS, and a PARTY may also be *on the other side of* one or more REPORTING RELATIONSHIPS.

This does not negate the value of also showing EMPLOYMENT as we did before, but it does allow us to represent any other relationship between two parties. The most important attributes of this entity are the "effective date" of a relationship, its "until date," and its reporting relationship "type," such as "organizational structure," "club membership," "family relationship," and the like. REPORTING RELATIONSHIP, then, is the fact that one PARTY is related to another at a particular time.

The power of this concept may be seen in many areas. For example, hospitals commonly band together into buying groups to obtain quantity discounts on purchases of pharmaceuticals and other hospital supplies. A buying group's blanket purchase order specifies a group discount, and allows each member hospital to issue a purchase order for items at that group price. To handle this arrangement, it is a simple matter to define the buying group as an ORGANIZATION, and identify a blanket purchase order for it, specifying the

prices. When a participating hospital's purchase order is received, it is necessary only to look up any buying group with which the hospital has established a REPORTING RELATIONSHIP. Once the contract price negotiated for that group has been found, it may then be applied to the purchase by the individual hospital.

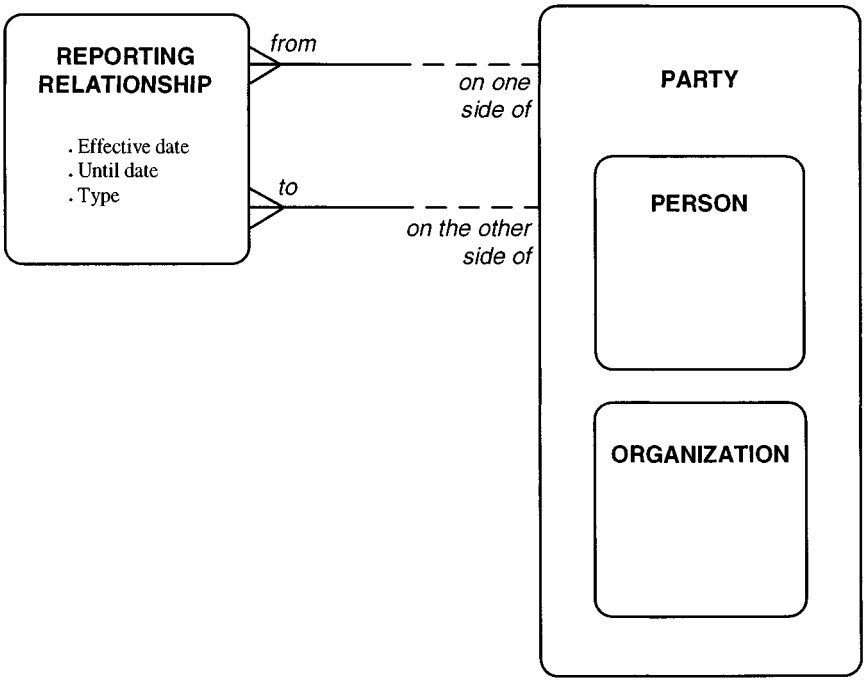


Figure 3.18: Reporting Relationships Between Parties.

REPORTING RELATIONSHIP allows *any* relationships among people and organizations to be defined. As mentioned above, the special case of people's relationships with their employers is elaborated in the entities POSITION ASSIGNMENT, TITLE, and POSITION. These entities will appear in many data models.

ABOUT TYPES

Note, that we have specified reporting relationship "type" as an attribute. Previously, EMPLOYMENT, SITE, GEOGRAPHIC LOCATION, and POSITION ASSIGNMENT also had "type" attributes. PARTY didn't show a "type" attribute, but it could have. In each case, presumably there is a finite list of possible values for the attribute. The "... type" attribute may be handled in one of three ways: If this list is relatively stable, it may be contained in a domain for the type. That is, the list of values for the attribute is documented in the data dictionary as a relatively

fixed list. Alternatively, if the list is also comparatively short, each of the "... types" could be shown as a subtype of REPORTING RELATIONSHIP.

If the list is more dynamic and variable, however, or if there is a reason to display the fact that such a list exists, it can be shown in the model as a new entity. This entity can be named REPORTING RELATIONSHIP TYPE, where each REPORTING RELATIONSHIP must be *an example of* one and only one REPORTING RELATIONSHIP TYPE, and each REPORTING RELATIONSHIP TYPE may be *embodied in* one or more REPORTING RELATIONSHIPS.

In short, it is possible to deal with REPORTING RELATIONSHIP TYPE (or any other ... TYPE) either as an attribute with a defined domain, as a set of subtypes, or as a relationship to a ... TYPE entity.

ABOUT POINTS OF VIEW

You will find in this book a bias toward creating the purest models possible, with an emphasis on describing things in terms abstract enough to encompass a wide range of circumstances. In the course of this chapter, however, we have discovered the purest model to be often in conflict with practical issues of addressing the perspectives of future systems users. (The case of ADDRESS and mailing lists is a good example.)

In real projects, however, you rarely are called upon to encompass a wide range of circumstances. Your client or user will have a particular problem to be addressed quickly and in terms that he or she understands. It may be unavoidable that you have to draw a model in those terms. So be it. The rent must be paid. Even when this happens, however, it is to your advantage at least to *understand* the more abstract model. You may even want to sketch it out on paper and file it, so you will have an answer if (when?) the client has a change of mind and a widening of perspective—immediately followed by the demand for you to deal with it ("I know that's what I said then, but this is *now*!").

IN SUMMARY

The first anchor for any data model is the entity PARTY, which encompasses the PEOPLE and ORGANIZATIONS of interest to the enterprise. By convention, we will put it along the right side of the model, along with such entities as PLACEMENT *at* a SITE, which is *in* a GEOGRAPHIC LOCATION, as well as EMPLOYMENT, which is *the basis for* a POSITION ASSIGNMENT *to* a POSITION. REPORTING RELATIONSHIP will lie alongside PARTY as well.

The second anchor for any data model is the stuff that the company uses, makes, and otherwise manipulates. That is the subject of the next chapter.

This page intentionally left blank

INDEX

NOTE: Page references to entity names are prefaced by *Enty.* References to attribute names are prefaced by *Attr.* References to pseudo-entities are prefaced by *Psu.* References to a main entry are indicated by a ~ symbol.