The Practical Guide to Business Process Reengineering Using IDEF0
Also Available from DORSET HOUSE PUBLISHING CO.

Complete Systems Analysis: The Workbook, the Textbook, the Answers
by James & Suzanne Robertson  foreword by Tom DeMarco

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

Data Model Patterns: Conventions of Thought
by David C. Hay  foreword by Richard Barker

Designing Quality Databases with IDEF1X Information Models
by Thomas A. Bruce  foreword by John A. Zachman

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

Quality Software Management Series  by Gerald M. Weinberg
Vol. 1: Systems Thinking
Vol. 2: First-Order Measurement
ISBN: 0-932633-24-2  Copyright ©1993  360 pages, hardcover
Vol. 3: Congruent Action
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

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
Dedication

I dedicate this book to Douglas T. Ross, the father of SADT and IDEF0, with whom I worked for more than thirty years. The concepts presented here are his. I have tried to present his ideas as I have come to understand and apply them to the real world. I apologize to him for any misleading statements herein, or for dabbing on his canvas, as he used to put it.
This page intentionally left blank
any different graphics software packages were used to produce the figures in this book. Thus, the reader will see several variations in style when reading the IDEF0 diagrams presented here. The use of a particular package to produce a specific form of IDEF0 diagram is not intended to imply endorsement by the author. In most cases, the package used to develop an IDEF0 model was determined by the product used by the client for whom the model was developed.

The following graphics packages were used to produce the diagrams in this book, listed in alphabetical order of the package name:

Authormate, by Eclectic Solutions, Inc.
AutoCad, by Autodesk, Inc.
BPwin, by Logic Works, Inc.
Design/IDEF, by Meta Software, Inc.
SA/BPR, by Popkin Software and Systems, Inc.
Contents

Figures & Tables  xii
Foreword  xvii
Preface  xix

Chapter 1: Introduction to the Method  1

Communication with IDEF0  1
Overview of IDEF0 Syntax  2
Origins and History of IDEF0  4
   An Early Opportunity  5
   A Good Name  6
   A Series of Methods  7
Elements of IDEF0  8
Use of IDEF0 with Other Methods and Tools  10
   IDEF0 and Scope of Application  10
   IDEF0 and the System Development Life Cycle  12
   IDEF0 and CASE  12
   IDEF0 and Systems Analysis  13
   IDEF Methods and Object-Oriented Methods  14
   IDEF Models and Integration with Other Methods  14
Chapter 2: Using IDEF0 for Process Improvement 15

Use of IDEF0 in Support of BPR 16
Use of IDEF0 for Military Downsizing and Reorganization 17
Benefits of Using IDEF0 Models for Enterprise Reengineering 20
The Reader/Author Cycle and the Workshop Approach 21
The AS-IS Model 24
  Case 1: Facilitate Information Flow 25
  Case 2: Reduce Costs 27
  Case 3: Analyze Performance 29
  Case 4: Introduce Business Process Concepts 33
Conclusions from Commercial and DoD BPR Efforts to Date 37

Chapter 3: Features and Benefits of the IDEF0 Activity Modeling Method 39

The Seven Basic Principles 39
Scope of Subject Matter That Can Be Handled by IDEF0 40
Benefits Resulting from the Use of IDEF0 41
Features of IDEF0 Analysis 43
Comparison to Data Flow Diagramming 45
Understanding a Top-Level IDEF0 Diagram of an Enterprise 49
Levels of Abstraction in IDEF0 Models 51
The Role of Data Analysis Compared to IDEF0 Function Analysis 53

Chapter 4: IDEF0 Graphic Language Syntax and Semantics 55

Activity Box Syntax 56
Arrow Syntax Elements 57
The IDEF0 Diagram Form 59
IDEFO Model Rules 63

Chapter 5: Pragmatics 81

Defining the Purpose, Viewpoint, and Type of Breakdown 82
Starting the Model 84
Validating the Model 88
Modeling Other Viewpoints 92
Using Node Diagrams, Schematics, and FEO Diagrams 95
Using the Reader/Author Review Process 101
Chapter 6: Do's and Don’ts  106

Language Rules  107
Box Rules  107
Arrow Rules  110
Diagram Rules  115
Model Rule  117
Model Quality Measurement  122
Achieving Model Quality  135
Management Lessons Learned  140

Appendices  145

Appendix A: Types of Models and Forms of Breakdown  147

Modeling the Function versus the Process  147
Forms of Activity Breakdown  149

Appendix B: Sample Models  156

Appendix C: SADT and IDEF0—A Historical Perspective  172

Comparison of SADT with IDEF0  173

Appendix D: SADT Data Models, IDEF0, and IDEF1X  175

Appendix E: A Modeling Process Case Study  179

Background of the Restaurant  179
Excerpt from an Interview with the President and CEO  180
Excerpt from an Interview with Jeff Shermon  182
Modeling Ben's Burgers  182

Key Terms  207

Index  213
FIGURES & TABLES

Figure 1-1: Simple Box and Arrow Graphics. 2
Figure 1-2: Simple Control Graphics. 3
Figure 1-3: Simple Mechanism Syntax. 3
Table 2.1. The DoD/CIM 25-Step BPR Process. 19
Figure 2-1: Keep Contract "Actuals" Records. 26
Figure 2-2: Organize & Manage the Effort. 28
Figure 2-3: Develop Company's Public Image. 30
Figure 2-4: Node Diagram of Hiring Model. 31
Figure 2-5: Be a Successful Group General Manager. 32
Figure 2-6: Run Successful Group Operations. 32
Figure 2-7: Establish Local Group Controls. 33
Figure 2-8: Evolve the Enterprise Over Time. 34
Figure 2-9: Run the Enterprise. 35
Figure 2-10: Manage the Enterprise. 36
Figure 3-1: Breakdown into Six or Fewer Activities. 43
Figure 3-2: Top-Down IDEF0 Model Structure. 44
Figure 3-3: A Data Flow Diagram to Track Assembly Status. 45
Figure 3-4: Same Track Assembly Status Data Flow Using IDEF0 Format. 46
Figure 3-5: Assembly Status Tracking IDEF0 Diagram, with Physical Parts Assembly Added. 47
Figure 3-6: Previous Diagram, with Controls Added. 48
Figure 3-7: Complete IDEF0 Diagram, with Mechanisms Added. 48
Figure 3-8: A Typical IDEF0 Diagram. 50
Figure 3-9: Mechanism at a Lower Level of Abstraction. 52
Figure 4-1: Activity Box Syntax. 56
Figure 4-2: Arrow Syntax. 58
Figure 4-3: Semantics of Arrow Forks and Joins. 59
Figure 4-4: The IDEF0 Diagram Form. 60
Figure 4-5: Diagram of the IDEF0 Modeling Process. 62
Figure 4-6: Model Structure of Diagrams. 63
Figure 4-7: A-0 Diagram with Purpose and Viewpoint. 64
Figure 4-8: Node Structure of a Model. 65
Figure 4-9: Sample Node Diagram. 66
Figure 4-10: ICOM Code Definition. 67
Figure 4-11: Remote Reference. 68
Figure 4-12: Reference Expression Illustrations. 69
Figure 4-13: Illustration of ICOM Coding to Connect Boundary Arrows. 70
Figure 4-14: Mechanism and Call Syntax. 71
Figure 4-15: Multiple Mechanisms. 72
Figure 4-16: Mechanism Arrow Branching. 73
Figure 4-17: Call Points to Further Detail. 74
Figure 4-18: Multiple Calls to a Common Model. 75
Figure 4-19: Default Values at Model Interfaces. 76
Figure 4-20: Call to Separate Process Models. 77
Figure 4-21: Examples of Mechanism and Call Combinations. 79
Figure 4-22: Examples of DRE and Call Arrow Use. 80
Table 5-1. IDEF0 Project Personnel. 84
Figure 5-1: Arrow Layout Rules. 87
Figure 5-2: A Highlighted Diagram. 90
Figure 5-3: A Publication Format Diagram. 91
Figure 5-4: Intermodel Tie Syntax. 92
Figure 5-5: Management Viewpoint Model. 93
Figure 5-6: Technical Viewpoint Model. 93
Figure 5-7: Call Arrow Illustration. 94
Figure 5-8: Sample Node Diagram for Communicating Model Content. 95
Figure 5-9: Process of Creating a Schematic. 96
Figure 5-10: FEO Showing First Librarian Event. 98
Figure 5-11: The Raveled-Rope FEO. 99
## FIGURES & TABLES

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 5-12:</td>
<td>Arrow Trace Process.</td>
<td>100</td>
</tr>
<tr>
<td>Figure 5-13:</td>
<td>The Reader/Author Cycle.</td>
<td>102</td>
</tr>
<tr>
<td>Figure 5-14:</td>
<td>The Reader Kit Cover Sheet.</td>
<td>103</td>
</tr>
<tr>
<td>Table 6-1.</td>
<td>Summary of IDEF0 Rules.</td>
<td>107</td>
</tr>
<tr>
<td>Figure 6-1:</td>
<td>Box Numbering Convention.</td>
<td>108</td>
</tr>
<tr>
<td>Figure 6-2:</td>
<td>Input, Output, and Control Conventions.</td>
<td>109</td>
</tr>
<tr>
<td>Figure 6-3:</td>
<td>Activity Naming Convention.</td>
<td>110</td>
</tr>
<tr>
<td>Figure 6-4:</td>
<td>Plug-in Alternative TO-BE Processes.</td>
<td>111</td>
</tr>
<tr>
<td>Figure 6-5:</td>
<td>Boundary Arrow Compatibility Requirement.</td>
<td>112</td>
</tr>
<tr>
<td>Figure 6-6:</td>
<td>Arrow Uniqueness Rule.</td>
<td>113</td>
</tr>
<tr>
<td>Figure 6-7:</td>
<td>Arrow Flow-Restriction Rule.</td>
<td>114</td>
</tr>
<tr>
<td>Figure 6-8:</td>
<td>Arrow Corners.</td>
<td>115</td>
</tr>
<tr>
<td>Figure 6-9:</td>
<td>Flow Chart and IDEF0 Diagram Conventions Compared.</td>
<td>117</td>
</tr>
<tr>
<td>Figure 6-10:</td>
<td>Manufacturing Diagram with Mixed Viewpoints.</td>
<td>118</td>
</tr>
<tr>
<td>Table 6-2.</td>
<td>Personnel, Their Activity Level, and Their Viewpoints.</td>
<td>119</td>
</tr>
<tr>
<td>Figure 6-11:</td>
<td>Manufacturing Diagram with Technical Viewpoint.</td>
<td>120</td>
</tr>
<tr>
<td>Figure 6-12:</td>
<td>Manufacturing Diagram with Quality Viewpoint.</td>
<td>120</td>
</tr>
<tr>
<td>Figure 6-13:</td>
<td>Manufacturing Diagram with Financial Viewpoint.</td>
<td>121</td>
</tr>
<tr>
<td>Table 6-3.</td>
<td>Definition of the Fog Factor.</td>
<td>124</td>
</tr>
<tr>
<td>Figure 6-14:</td>
<td>Comparison of Arrow Labels.</td>
<td>125</td>
</tr>
<tr>
<td>Figure 6-15:</td>
<td>Comparing Names and Labels.</td>
<td>126</td>
</tr>
<tr>
<td>Figure 6-16:</td>
<td>A Too-Simple Diagram.</td>
<td>127</td>
</tr>
<tr>
<td>Figure 6-17:</td>
<td>Revised Diagram.</td>
<td>129</td>
</tr>
<tr>
<td>Figure 6-18:</td>
<td>Lack of Arrow Bundling.</td>
<td>130</td>
</tr>
<tr>
<td>Figure 6-19:</td>
<td>Equivalent Diagram with Arrow Bundling.</td>
<td>131</td>
</tr>
<tr>
<td>Table 6-4.</td>
<td>Common Labels for Control-Oriented Arrows.</td>
<td>132</td>
</tr>
<tr>
<td>Table 6-5.</td>
<td>Common Mistakes and Their Solutions.</td>
<td>134</td>
</tr>
<tr>
<td>Figure 6-20:</td>
<td>Content-Free Diagram.</td>
<td>136</td>
</tr>
<tr>
<td>Figure 6-21:</td>
<td>Clarified Diagram.</td>
<td>137</td>
</tr>
<tr>
<td>Figure 6-22:</td>
<td>First Attempt at an ERDA Diagram.</td>
<td>138</td>
</tr>
<tr>
<td>Figure 6-23:</td>
<td>Function-Oriented Improved Diagram.</td>
<td>139</td>
</tr>
<tr>
<td>Figure A-1:</td>
<td>Process versus Function Modeling.</td>
<td>148</td>
</tr>
<tr>
<td>Figure A-2:</td>
<td>Decomposition Example.</td>
<td>150</td>
</tr>
<tr>
<td>Figure A-3:</td>
<td>Sequential Example.</td>
<td>151</td>
</tr>
</tbody>
</table>
Figure E-16: Operate Ben's Burgers. 196
Figure E-17: Operate Ben’s Burgers. 197
Figure E-18: Operate Ben’s Burgers. 198
Figure E-19: Operate Ben’s Burgers. 200
Figure E-20: Manage Local Restaurant. 201
Figure E-21: Order Supplies. 202
Figure E-22: Step 6, Run Scenarios. 203
Figure E-23: Step 7, Run the Reader/Author Cycle. 204
Over the past several years as I have traveled just about everywhere to discuss business reengineering, it has amazed me that most people think reengineering is a new phenomenon created by Mike Hammer. Some have been dumbfounded when I say how long it has been around and what it has accomplished during that time. I certainly give credit to Dr. Hammer’s book for the publicity reengineering has received, but it is the unsung heroes that have made reengineering not just well known but successful.

One such hero is Clare Feldmann. He and I have been talking about and through his book for at least five years. Clare is one of those guys who prefers the background to the limelight. His years at SofTech with Doug Ross were the tough years as reengineering emerged into a modern method. More importantly, Clare was one of the pioneers in the belief that understanding processes and activities is the key to the application of technology to problems. This book shares with you many of his thoughts on how to effectively reengineer.

I will share one personal insight for you to bear in mind as you read the chapters that follow. Over the years, Clare and I have shared a number of opportunities, committees, boards, and so forth. At each, he takes very careful notes. In fact, more often than not, he converts them to IDEF0 activity diagrams and data diagrams. He has always claimed that he understands subjects better through models. By this time, he must have thousands of them. While I have no secret knowledge here, I will bet that a model exists for how to write a book. Ask him when you meet him.
Clare's love of models led to a deep understanding of how processes work. A lot of what we now say and do in reengineering efforts comes from that approach: Understand what you do now and look at what you can do to improve the process. When you model the reengineered process, it quickly becomes obvious how technology can help to further optimize the efficiency of the new process. All of this is quite logical—yet completely impossible to achieve without models. Clare's favorite technique is shown here—IDEFO—the integrated definition language that he helped create in the 1970's. That modeling technique captures information not obtainable otherwise. Understanding the controls on actions as well as the mechanisms that facilitate action is critical to creating business rules that clarify action.

Another benefit of Clare’s effort is the timing. For those of us in Government, rapid change is upon us. The Information Technology Management Reform Act of 1996 now mandates that managers redesign processes before recommending information technology investments. In fact, this is a first step in bringing accountability to the hundreds of billions of dollars expended each and every year on technology in Government. With IDEFO, Clare shows managers a proven way to assure that those investments make sense.

Any time someone writes a book on reengineering, I cheer. Particularly so when the person writing the book has gotten his hands dirty with real work, in helping managers improve. We have too many volumes out there that discuss others' efforts, written by people who have never tackled a project themselves. Here you have a master who is describing his art—and the only charge for his services is the cost of the book. What a bargain!

If you only have time to read one book on the subject, this is it. Enjoy.

John V. Tieso

Former Deputy Director, Office of Functional Process Improvement
Office of the Assistant Secretary of Defense (OASD) for Command and Control, C3I
Arlington, Virginia
For many years, there has been a call for a comprehensive book on the IDEFO method and its use in BPR. This book, written in answer to that call, is designed to introduce the method, its usage, and the benefits that can be achieved. It is also written to serve as a “do’s and don’ts” document to identify the ways the method has been misused in the past and to specify ways to correct these misuses.

This book presents essential information about the IDEFO method—its definition, basic rules of usage (including the standard language syntax and semantics as contained in the Federal Standard)—and lessons learned from many years of application in the real world. The examples depict actual commercial client and government agency models, from which names and proprietary information have been removed.

**Who Should Read This Book**

The material presented in this book is suitable for three categories of IDEFO users:

- people who wish to explore the concepts and the application potential of IDEFO for enterprise reengineering
- people who wish to determine how IDEFO might be helpful to them in their systems analysis efforts
• people who have been introduced to IDEF0 concepts, and who wish to understand the theory behind the concepts so that they can use the method more effectively.

The Department of Defense (DoD) Corporate Information Management (CIM) Office reports that more than 600,000 people worldwide fall into the third category, and one goal of this book is to provide meaningful assistance to such people during their IDEF0 analysis efforts.

After reading the book, a person should be familiar enough with IDEF0 to read and understand any IDEF0 model and should be able to understand how the method might be applied to enterprise or systems analysis and what goals and benefits are reasonable to expect from its application. However, to be considered an "author" of IDEF0 models, he must complete additional formal training and a brief apprenticeship period with an experienced IDEF0 author.¹

Training courses in IDEF0 modeling are available from commercial vendors and at various universities, primarily in the United States;² SADT training courses are available worldwide.³ In addition, there are more than a dozen good computer tools that support the IDEF0 method.

My hope is that this book will guide a new generation of modelers in producing the maximum benefit from IDEF0.

Winchester, Massachusetts
April 1998

C.G.F.

¹ Throughout this book, the male pronoun is used generically to include both males and females. Also, please note that in IDEF0 terminology, a reader is someone who knows the rules of the IDEF0 graphics and can therefore understand the diagrams. An author is someone who is a reader but who also is trained in creating IDEF0 models.

² For a list of registered vendors of courses and computer support tools for the IDEF family, write to Society for Enterprise Engineering, ATTN: Secretariat, c/o The Koop Foundation, 15825 Shady Grove Road, Suite 22, Rockville, MD 20850.

³ SADT is the precursor to IDEF0. For more on SADT and the history of IDEF0, see Chapter 1 and Appendix C.
The Practical Guide to Business Process Reengineering Using IDEF0
This page intentionally left blank
This chapter presents the basic principles of IDEFO—the assumptions behind its graphics and the rules for applying its analytical process. Expanding on Chapter 1, the discussion examines the scope of topics that can be handled by IDEFO, the benefits of its use, and a comparison to other popular graphical diagramming methods. Chapter 4 gets even more specific as it presents the detailed syntax and semantics of the IDEFO graphical language, as officially defined in the Federal Standard.

**The Seven Basic Principles**

There are seven principles—or rules of conduct—that must be followed in order to apply IDEFO techniques successfully:

1. **The method must accurately represent the problem area:** A graphical model of a system should be developed so that the system elements and their interactions can be defined, documented, communicated, discussed, and analyzed effectively.

2. **The model must have a top-down, modular, hierarchical structure:** The model should depict the system top-down by defining modular system elements that interact to form a hierarchy.

3. **The model must separate function from design:** By definition, what the system does (function) must be kept separate from how it does it.
(design). That is, more than one design can be developed for a single function; this distinction is important. Because of this, design improvements may be developed without disturbing the basic function. If a more drastic (functional) change is called for, then basic functions can be changed and a new design created to meet the new functional requirements.

4. **The model must reflect both objects and actions—things and happenings:** The modeling method must be able to depict all forms of things and happenings. Restriction of the modeling scope to raw data would leave out the people, resources, raw materials, human factors, and other important influences on the operation of the enterprise, all of which are critical in managing change.

5. **The model form must be graphical:** The form on which the model is recorded must be graphical, not mathematical or textual. The graphical form must communicate concisely and rigorously to the people who must validate that the model reflects the real functions and processes of the enterprise.

6. **The model must be the product of disciplined, coordinated teamwork:** To build a model and to achieve consensus among the enterprise staff requires disciplined, coordinated teamwork. The IDEFO method must therefore contain the working rules and procedures for developing and validating the model in an organized way (reader/author cycle forms and procedures, librarian configuration management procedures, and the like).

7. **The model must present all information in writing:** By having a standardized form on which all information (interview notes, conclusions with their rationale, definitions of terms, and so on) is retained, the method assures that information is not lost during the early stages of the planning cycle. With its forms and procedures, IDEFO provides a convenient means of retaining information.

**Scope of Subject Matter That Can Be Handled by IDEFO**

As we have stated, IDEFO is a method for analyzing enterprises. An enterprise may be a company, a division of a company, a group of companies working together in a joint venture, a field office of a parent company, a support function within an office (for example, publications, accounting, or payroll), a government agency (for example, the Department of Defense or the Commerce Department), an individual running a small business, or a project within a company. All of these are
enterprises, since they are composed of one or more people working to accomplish some form of product development or to provide some service.

The scope of the analysis encompasses manual as well as computerized aspects of the enterprise. IDEFO is particularly useful for rigorously defining the juncture at which manual actions interface with the computer (the so-called man-machine interface). For example, using the viewpoint concept of IDEFO, the author may develop a model from the user's viewpoint as well as from the software developer's viewpoint, thus providing a requirements definition from both important aspects of a software system to be developed. The model bridges the communication gap between user and software cultures.

IDEFO analysis does not stop at the man-machine interface, but may be used to model the operation of the software system itself. In fact, the method has been successfully used for many years to model the requirements, top-level design, and detailed design of software systems.

**Benefits Resulting from the Use of IDEFO**

The primary goal of IDEFO is to facilitate change management, a goal that was introduced in Chapter 2, under the topic "Use of IDEFO for Military Downsizing and Reorganization." In this chapter, we discuss specific benefits that can be expected from the IDEFO modeling effort.

Stated simply, the goal of the modeling effort must not stop at developing a model. Such a goal leads to models that sit on a shelf and do not achieve real operational benefits. Instead, the purpose of an IDEFO model should be "to provide a baseline on which to plan and manage change." The model serves as a communication vehicle for streamlining existing processes, for introducing new technology, for evaluating the impact of a change on related processes, and for achieving consensus among the staff involved in the change.

The initial benefits are achieved by the analyst as he creates (authors) the model. In developing the model, the analyst cannot help but identify problems and potential improvements in the operation of the enterprise. When the analysis is complete, the paper model is available, thereby providing documentation on-the-fly and not requiring a separate, after-the-fact documentation phase at the end of the analysis. Specific benefits of IDEFO modeling are as follows:

*Identifies Needs:* A good IDEFO model will look very simple and easy to understand. Needs and opportunities for improvement revealed during the modeling effort may seem obvious at first, but they would have gone unnoticed otherwise.
Builds Consensus: The diversity of a staff's background, training, skills, and knowledge can impair communication. The reader/author model critique method can establish a well-defined common basis for understanding by the entire staff. Real consensus, not just imagined consensus, can be reached.

Enhances Vision: The condensed, graphical picture provided by the model presents a thought-provoking baseline on which to consider improvements. An analyst or an enterprise planner who considers potential use of a new technology or new method in light of this baseline may identify specific application opportunities.

Provides a Basis for an Open Architecture: The model structure may be used to define the interfaces between system elements, and to identify precise interfaces for the definition of an open systems architecture. The model bounds the scope precisely by showing where the modeled system fits into the bigger picture.

Broadens Automation and Commonality Potential: Almost any model of any size will reveal activities that are similar to those already modeled in other parts of the structure. These similarities can be used to define improvements, such as a subsystem to support these common activities.

Supports Management Control Through Metrics: The model may serve as a baseline for cost, time, flow capacity, peak loads, and other metrics relevant to the design of a new system. If the model detail is not sufficient to attach specific metrics, the lack of detail should be used to identify where further decomposition is needed. Finer detail in the modeling will facilitate more precise measurement.

Defines Variants for Broader Support System Use: Support systems that are applied to several activities in the model may benefit from a careful analysis of the actual functionality needed at each point of usage. Variants and versions of the support system could save considerable cost if the support system can be fine-tuned for each specific usage point.
Features of IDEF0 Analysis

The IDEF0 method analyzes a subject by investigating how it operates, by modeling the operational details uncovered, and by applying evaluation techniques (such as "value added" criteria) to identify potential improvements. The author of the model uses two-dimensional, simple graphics as a means of capturing and communicating the results of the analysis.

Applying IDEF0 to the analysis of an enterprise thus results in a paper, two-dimensional graphical model of the enterprise. The model is hierarchical in nature, consisting of a set of diagrams in which each diagram represents three to six activities of the enterprise. At the top of the IDEF0 model, the enterprise is decomposed into three to six very large activities or subsystems that represent the activities of the enterprise (see Figure 3-1). These activities are shown as rectangular boxes.

![Diagram](image)

Figure 3-1: Breakdown into Six or Fewer Activities.

Once the top-level activity breakdown has reached consensus among the enterprise staff, each of the boxes is further broken down into the next level of detail on a new three-to-six activity diagram. Thus, the entire IDEF0 model provides a tree
structure of diagrams. Anyone wishing to inspect a particular aspect of the enterprise may scan the tree structure, identify the activity and level of detail that is useful for his purposes, and then retrieve the corresponding IDEF0 diagram to study the complete description of the selected activity at the selected level of detail. Figure 3-2 shows a graphical representation of an IDEF0 model that breaks down an enterprise into three levels of detail.

Figure 3-2: Top-Down IDEF0 Model Structure.

At the top level, the entire system being modeled (labeled Whole System in the figure) is shown as broken down into four parts, labeled 1, 2, 3, and 4. Part 2 is further broken down into a three-part diagram, shown below the initial diagram as emanating from a shaded box 2. Finally, boxes 1 and 3 of the detailed part 2 box are broken down into yet finer detail in the two diagrams displayed at the bottom.
of the figure. This illustrates an important feature of the IDEFO model: Any aspect may be broken down to be as detailed as necessary, while other aspects may be less completely decomposed. This permits the analyst to zero in on the most critical aspects to be analyzed, while putting off the other aspects until needed.

Comparison to Data Flow Diagramming

Most popular in past decades, the data flow diagram approach to modeling a computer software system is still used today for software design. It provides a picture of the data that are input to and output from the software. Figure 3-3 shows that Location of Assembly and Quality Measurements are input to a Scheduling Database of information about the status of a part. When a process has been completed, the assembly machine sends a “Process Complete” Status signal to the database, as an additional input. Thus, the database has a running record of where the part is located and which processes have been performed. As output, the software produces a report on the status of the part.

![Figure 3-3: A Data Flow Diagram to Track Assembly Status.](image-url)
Figure 3-4 shows the same information as Figure 3-3, but this time as an IDEFO diagram.

Equivalent Information to the Data Flow Diagram

Figure 3-4: Same Track Assembly Status Data Flow Using IDEFO Format.

Note that the interfaces between the activity boxes in the IDEFO diagram are represented by arrows, depicting one activity supplying input to or controlling one or more other activities (as we shall see in Figures 3-5 through 3-7). Unlike the simpler data flow diagram, the interactivity relationship can include data flow, raw material flow, management and other forms of control, resources (such as people, machines, and computers), and whatever else must be included to model the complete enterprise's operational details.

Figures 3-5 through 3-7 detail information that is not shown on the data flow diagram. Whereas a data flow diagram is adequate to depict information needed by a computer software program, it is not adequate for reengineering purposes, since there are other factors influencing an enterprise's operations. These other factors must be included if the reengineering process is to be understood and designed, and the changeover process controlled. The IDEFO model shows what controls each activity and who performs it, as well as the resources needed by each activity.
Figure 3-5 shows the addition of the physical part proceeding through the stages of manufacturing, from Unassembled Part entering box 1 to Completed Part leaving box 3.

Addition of the Physical Product Being Processed

Figure 3-5: Assembly Status Tracking IDEFO Diagram, with Physical Parts Assembly Added.

Figure 3-6 adds controls to each IDEFO activity box. For example, we see that there is an Assembly Schedule that controls when the part is prepared for processing. Company Manuals are used to control the remaining three activities. These controls must be carefully considered when making changes, and therefore are important to include in the reengineering effort.

Finally, Figure 3-7 shows what mechanism performs each process. This includes people (for example, Test Engineers for box 3) as well as support tools and systems (for example, Testing Facilities for box 3). A common practice is to document the elements of the enterprise's organization chart as mechanisms in the model in order to show the assignment of responsibility to specific organizational elements. In the TO-BE picture, the revised organizational responsibility is likewise shown.
Addition of Control Information -- What Controls Each Activity

Figure 3-6: Previous Diagram, with Controls Added.

Addition of Mechanisms -- What Performs Each Activity

Figure 3-7: Complete IDEF0 Diagram, with Mechanisms Added.
An important difference between data flow diagrams and IDEF0 diagrams is the scope of the topic covered: DFDs were intended to depict fine detail needed by a software design specification, whereas IDEF0 diagrams were designed to capture very complex topics such as the operation of the aerospace industry, a government agency, or a business.

IDEFO syntax may be used to capture fine detail, as we have just shown in the series of diagrams comparing the information content of DFDs to IDEFO diagrams. However, many of the features of IDEFO have the objective of handling great complexity, not depicting fine detail. The purpose and viewpoint statements of an IDEFO model are examples of features that enable handling complexity.

While the purpose of a set of data flow diagrams is to depict the operation of computer software, the purpose of an IDEFO model may be one of many, since the modeling method is applicable to any "system" comprised of "things" and "happenings." Focusing the analysis effort by purpose shortens the list of enterprise elements that must be modeled during a reengineering effort. In other words, the analyst must narrow the topic to be modeled and decrease complexity.

An IDEFO model must also select a specific viewpoint. If all viewpoints of an enterprise were to be included in a single model, the information would be too complex to understand.

**Understanding a Top-Level IDEFO Diagram of an Enterprise**

Figure 3-8 presents a typical top-level diagram in an IDEFO model (it originally appeared as Figure 2-9). This diagram might represent the most general level of detail of a model entitled "Run the Enterprise."

The figure shows the running of an enterprise, broken down into five major functions: 1) Manage the Enterprise; 2) Manage Day-to-Day Operations; 3) Manufacture Products; 4) Provide Management & Technical Information; and 5) Provide Resources. Four of these major functions are shown as boxes positioned in a stairstep fashion that makes the control and feedback loops between the functions simpler to draw. The fifth box (Provide Resources) is shown below the stairstep, since it is a support function that provides resources (people and machines) to be used by the other functions.

By reading the names in the boxes, anyone can get a quick idea of the breakdown of the subject Run the Enterprise, as well as the topics to be modeled in subsequent lower-level diagrams. For example, if the reader wishes to examine further detail regarding the provisioning of resources, he would isolate the diagram for box 5 and study that at a finer level of detail.
IDEF0 graphics are undeniably straightforward and that quality is, in my view, one key to their success. Such elaborations as additional box shapes, dotted arrows, and formalisms for arrow branching structures have been experimented with by modelers since the inception of IDEF0. It seems only natural to conclude that such elaborations would provide additional expressive capabilities to the analyst.

However, I have learned to resist the temptation to elaborate the syntax, since this makes the graphics too complex to fulfill the essential role of the model to communicate. At the early stages of project planning, communication is the key element, and the simple graphics that are quickly learned and employed must remain a basic element of IDEF0.
Levels of Abstraction in IDEFO Models

As stated, the breakdown approach used in IDEFO modeling decomposes selected activities into finer and finer levels of detail. The resultant model set includes sufficient detail to plan and control the implementation of changes to the enterprise.

An IDEFO model may also contain multiple levels of abstraction. To understand what is meant by "level of abstraction," let's consider a typical enterprise. Each level of management has its natural level of abstraction. Top management must think in broad terms, taking into account many viewpoints as well as abstractions of the details of what is going on in the enterprise. An upper-level manager cannot possibly understand every detail of all of the activities in the entire enterprise, but he must abstract business processes and characteristics that are important at his level of abstraction. He must consider such things as overall profit, hiring policies, or the quality of the enterprise's product line, rather than the specific factors that go into these high-level abstractions.

Conversely, a manager at a lower level needs fine-level detail about a specific topic from a single viewpoint, and he may get uneasy thinking in general terms. His level of abstraction is concerned with such things as the cost of raw materials, the capabilities of specific staff members, or the quality of the service for which he is responsible.

The model's level of abstraction is independent of its level of detail. That is, after selecting a level of abstraction to model, the author may decompose the activities to whatever level of detail he requires to satisfy the purpose of the model. However, if the purpose of the model requires information at a different level of abstraction, no amount of further detailing at the present level of abstraction will provide the needed information. Modeling at all required levels of abstraction must be completed to satisfy the enterprise analysis requirements.

The IDEFO concept of a mechanism is different from the concept of level of abstraction. In IDEFO, a mechanism represents who performs the activity and what tools (software packages and equipment) are required to perform the activity. In other words, the mechanism identifies the resources needed to perform the function. A mechanism is therefore at a lower level of abstraction than the activity box it implements.

The fact that the mechanism depicts a lower level of abstraction does not mean that it cannot be modeled in IDEFO. It just means that the mechanism is shown as an arrow entering the bottom of the box, and that a separate model must be examined to understand how the mechanism works (see Figure 3-9).

The IDEFO diagram in Figure 3-9 shows Activity A23 (Edit Document) using the word processor support mechanism. The IDEFO model for the word processor is shown at the lower-right corner of Figure 3-9. What this tells us is that any word
processor mechanism may be modeled and plugged into the diagram to show the precise activity performed when editing a document with any variety of word processors. This shows how the editing is done using the mechanism, and therefore indicates that there is a drop in level of mechanization from what is done to the document by the EDIT activity.

Figure 3-9: Mechanism at a Lower Level of Abstraction.

To accomplish the reengineering of an enterprise, an analyst may need to show multiple levels of abstraction and several levels of mechanism, as well as how those are integrated to perform the processes of the enterprise. Showing all this provides a clear, accurate big picture of the overall operation of the enterprise.

In addition to an IDEF0 process-oriented analysis, various additional analysis methods may be applied, using the IDEF0 model as a baseline. For example, the analyst may need to know where costs and labor are being expended, in order to estimate the return on investment of potential changes or to understand the impact of introducing new technology. Various analyses, such as activity-based costing (ABC) and work flow simulation, are typically applied using the IDEF0 model as a baseline.
The Role of Data Analysis Compared to IDEF0 Function Analysis

Detail needed regarding the processes requires not only IDEF0 process models, but also precise information about the arrow content. This requires a careful analysis of data.

An analyst may be confronted with a tidal wave of raw data when he first gathers information about an enterprise. He typically starts by finding out the precise meaning of the terms that are key to the system's operation; then he documents these definitions. Once he has done that, he has a solid basis for initiating discussion with the staff about the detailed operation of the system. But even before analyzing the system in detail, he must consider the system functions and their interactions to understand the need, the purpose, and the objective of the relevant data.

Not all analysts take this view. In fact, some prominent leaders in the database community have stated that there is no need to look at the functions and activities at all. One database proponent wrote that the development of a function model actually gets in the way of the programmer and should be avoided if possible, further proclaiming his First Rule of CASE: "Only one diagramming method is needed—the data model—and code can be generated automatically from a data model."  

This statement is true if the automated system to be built performs simple, well-known processes such as basic banking transactions (for example, put money in checking account, withdraw money from checking account, transfer money to savings account, and so on). These activities are so familiar that an IDEF0 model would not be helpful, and design of the data structure of the system can be started almost immediately. The statement is not true, however, if the software to be built has a complex processing aspect, such as that for an avionics system or a computer's operating system.

The field of automated programming would also lead one to believe that analysis, design, and programming are outmoded—as if the system will do all this for you and produce a running computer program. However, the type of system that is implied here is one that simply queries a database and displays answers on a screen. Automatic programming has a long way to go before it can build an avionics system, a computer operating system, or a compiler. And haven't we slipped into the solution domain? Automated programming tools are of no help if you are trying to reengineer a business operation and have not yet understood what the system will look like.

Regardless of the system to be analyzed and the comparative importance of the data versus the activity analysis, at some point the data must be analyzed. To perform this portion of the analysis, the analyst must understand several important characteristics of the data.

Although there are many data characteristics that could be analyzed, two are critical: data dynamics and data structure. Data dynamics can be depicted by SADT data models, using a portion of SADT that was not adopted by the Air Force for IDEF0. Data structure may be modeled using IDEF1X, where the term "structure" signifies all non-dynamic data information, such as its attributes and its relationship to other data elements.

A type of function model, the SADT data model shows the breakdown of the kinds of data in the system, with arrows indicating which activities produce which data elements and which activities use the data for what purpose. This gives a dynamic picture of the data being created, used, and merged with other data to form new data entities. The SADT data model is a natural companion for an IDEF0 or SADT activity model, and it is useful for analyzing how the system manipulates data and what is affected when changes are implemented. It also provides a validation of the activity model. In fact, most data models developed using SADT result in corrections to the activity model!

The IDEF1X model provides all of the structural information about the data. Ultimately, relating the need for and the usage of specific attributes of data by the activity boxes of an IDEF0 model is the key to integrating the two models.
Accounting system examples, 25–27, 37–38
Activity, 56, 83–84
   See also Box; Breakdowns
clusters, 188–90, 208–9
defined, 207
layout of, 86, 191
list, 86, 187–90
maximum number of, 9, 86, 115–16
naming convention, 110
quality of, 123
rules for, 107–10
Activity-based costing (ABC) approach, 52, 185
Aerospace industry, 4, 5, 17
Air Force
   AFCAM program, 5, 6, 172
   computer-aided manufacturing and, 5
   ICAM program and, 10, 17
   IDEF0 and, 7, 17
   ManTech Office, 4, 5, 7, 173
   modeling method, 7
Air Force Computer-Aided Manufacturing
   (AFCAM) program, 5, 6, 172
Approval levels
   draft, 84, 101
   publication, 84, 101, 102, 104
recommended, 101, 102, 134
working, 84, 101, 103
Arrow, 2, 12, 55, 56, 57, 58, 86–87, 207
   See also Boundary arrow; Call arrow;
   Control arrow; IDEF0 rules; Input
   arrow; Mechanism arrow; Output
   arrow
branches, 58, 72, 114
bundling, 98, 107, 129–31, 134, 193, 208
content, 56, 57, 98–99, 114, 116
data interface, 30
defined, 207
draft, 84, 101, 103
external, 210
flow restriction rule, 114
fog factor and, 124
forced, 203
forks, 59, 62, 87
joins, 59, 62, 87
labeling, 56, 57, 87–88, 107, 113, 124–26,
   130, 131, 135
layout rules, 87
parentheses and, 112
pathways, 86, 87
pipeline, 59, 88, 203
plug-compatibility, 110–12
purpose of, 133, 134
railroad-track, 130
raveled-rope, 63
shape of, 57–58, 107, 114–15
syntax, 2, 12, 57–59
tracing, 100–101
benefit of, 24ff.
in BPR process, 19, 21, 23
CASE and, 13
defined, 208
DoD and, 18, 19
Assembly code, 4
Author, 22, 24ff., 82, 83, 84, 88, 204
See also Authoring steps; Reader/author review cycle
cost of development and, 141–42
defined, 207
labeling by, 87–88
lead author, 84, 85
mistakes of, 24ff., 134
number of, 84
project scope and, 33
role of, 22, 43, 84–88, 102, 103–5, 128
skills of, 140–41
training for, xx, 10, 13
Authoring steps, 13, 182
for Ben’s Burgers case study, 182
Automatically programmed tool (APT), 5
A-O diagram, 65, 142
A-O diagram, 57, 64, 85, 133, 207
A-O statement, 134
Block diagram method, 11, 23
Boeing, 5
Boundary arrow, 57, 69, 70, 107, 110–12, 203
defined, 208
parent diagram and, 203
Box, 2, 4, 11, 12, 43, 55, 56, 107–10, 208
See also IDEF0 rules
as action, 2
controls and, 56, 108–9
decomposition and, 110
defined, 208
dominance and, 191
inputs and, 56
layout of, 191
mechanisms and, 56
naming, 56, 59, 107, 109–10, 124–26
numbering convention, 56, 64, 107–8
outputs and, 56, 108–9
parent, 69–70, 86
rules for, 107–10
stairstep position, 49
syntax, 12, 56–57
three-six rule for, 9, 59, 61, 116
Breakage, 16, 20, 23
Breakdown of activities, 9–10, 43–44, 82, 83
See also Decomposition
data models and, 175ff.
forms of, 147–55
organizational, 152
partial SBT, 154–55
purpose of, 82
sequential, 150–51
split-by-type (SBT), 76, 153–54
type of, 82–84
viewpoint of, 82
Bundling, see Arrow bundling; Data bundling
Business process reengineering (BPR), xix, 12, 15, 19, 20ff.
arrows, role of, and, 133
AS-IS model in, 19, 21, 23
business planning and, 19, 25–27
cost of development and, 141–42
definition, 208
deliverable of, 23
Hammer and Champy approach, 15, 21
inductive reasoning and, 15, 21
job loss, threat of, and, 141
management and, 37–38, 140–43
measurement of, 23, 33
reuse of models and, 142
steps in, DoD/CIM, 19
Business rule, 133, 177
as constraint, 133
Call arrow, 56, 67, 71, 73–76, 94, 155, 158
DRE vs., 79–80, 118
mechanism arrow vs., 78–79
subroutine and, 77–78
Call point, 74
Case study
Ben’s Burgers, 179ff.
in process improvement, 25ff.
INDEX 215

Champy, Michael, 15, 21
Change implementation, 16, 20, 23
Change planning, 34
Child diagram, 69–70, 208
Cluster, see Activity cluster
C-number, 57, 61, 80, 89, 104
Commenter role in IDEF0 project, 84
Computer-aided software engineering (CASE), 10, 12–13
Configuration management in government bid effort, 27–29
model, 156, 166–71
Constraint, 11, 133
Control arrow, 2–4, 11, 47, 48, 56, 62, 88, 132–33, 204
cost constraint vs., 133
convention for, 108–9
identification of, 132–33
syntax, 56
Corporate Information Management (CIM) Policy Committee, xx, 16, 17, 19, 173
DoD/CIM 25-Step BPR Process, 19, 21
Data
arrows, 2
bundles, 189–90
list, 85–86, 142, 155, 187–90, 193
function-oriented, 139
levels of, 85
name of, 60
number of, 60–61
parent, 64, 89, 203, 210
production, vii
readability, 114
reader kit and, 104, 105, 203
rules for, 55ff., 107, 115–17
structure, 44, 55, 63
three-six rule for, 43–44, 59, 61
top-level, 21, 23, 49–50, 156–60
Divide-and-conquer approach, 5, 9–10
Documentation, 16, 22, 41, 85
Dominance of boxes, 191
Downsizing, 15ff., 135–37
Egoless environment, 14
Enterprise, 14, 40–41, 156–60, 179–205, 209
model of, 3, 13, 20, 29, 34, 35, 36, 156ff., 160–65
Enterprise engineering, see Business process reengineering
ERDA diagram, 138
Expert, 22, 37, 84, 204

OASD and, 15
steps of BPR process, 19, 20, 21, 132–33
Detailed reference process (DRE), 67–69, 80, 94, 112, 209
call arrow vs., 79–80, 118
syntax for, 68, 94
viewpoints and, 119
Diagram, 2, 11, 13, 95
See also A0 diagram; A-0 diagram; For-exposition-only (FEO) diagram;
Node diagram; Schematic diagram;
Text
activities, number of, 115–16
authoring steps, 182–205
balance, 86, 158, 159
child, 69–70, 208
constraint and, 107
content-free, 135, 136ff.
data flow diagram vs., 2, 10, 46, 107, 116
as decomposition, 64
as documentation, 23
flow charts vs., 10, 11, 107, 116–17
form, 55, 57, 59–63, 85
function-oriented, 139

see also Breakdown
defined, 209
diagram, 61, 65
diagram as, 64
example, 150
importance of, 88–89, 135
interfaces of, 69
sequence, stopping a, 143
in TQM effort, 35

Department of Defense (DoD), xx, 15ff., 21
budget and, 16, 18
CIM office and, xx, 17, 19
defined, 209
hospital example, 136–37
IDEF0 choice, 172

Downsizing, 15ff., 135–37

see Business process reengineering
Federal Information Processing Standard (FIPS) PUB 183, 9, 12, 55
input arrows and, 133
Finkelstein, Clive, 53
Fog factor testing, 124
For-exposition-only (FEO) diagram, 62, 63, 82, 95, 97–101, 173, 210
defined, 210
of librarian event, 97–98
raveled-rope, 99, 101
Function modeling, 11, 54, 55, 148, 207
analysis method, 7–8
Gane and Sarson design method, 12
Glossary, see Model glossary
Hammer, Michael, xvii, 15, 21
IDEF methods
series of, 7–8
Users Group, 10, 17, 37
IDEF1X method, 7, 54, 172, 175–78, 210
IDEF0 method, xix, xx, 1ff., 4–8, 10–15, 23, 25, 39–54
See also Activity; Arrow; Box; Diagram;
ICOM codes; IDEF0 rules; Model;
Modeling; Project; Pragmatics; Text
benefits of, xix, 4, 6, 11, 12, 18, 20ff., 23, 25ff., 41–42
Ben’s Burgers example, 179–206
BPR and, xix, 10, 16–17, 20–21, 23, 25ff.
CASE and, 10, 11–13
change management and, 41
communication and, 1ff., 15, 16, 20
concepts of, xx, 8, 10–15
configuration-management procedure, 101
costs and, 23, 25, 27–29
criteria for, 16–17
data analysis vs., 53–54
data flow diagramming vs., 2, 10, 45–49
DoD and, xx, 172
do’s and don’ts, 106–44
elements of, 8ff.
features of, 43–45
features excluded, 61, 173
field office example of, 29–33
graphical elements of, 2, 17, 20, 39
history of, 4–8, 11, 17–19, 172–74
IDEF1 and, 7, 172
IDEF1X and, 54, 172, 175–78
IDEF3 and, 7
IDEF2 and, 7, 172
information flow and, 25–27
language of, 8
lessons from use, 13–14, 140–43
life cycle and, 10, 11, 12ff.
management and, 25ff., 37–38, 140–43
methodology and, 7–8, 101
misuses of, xix, 10ff., 106ff.
object orientation and, 12, 14, 174
principles of, 8, 39–40
process improvement and, 15–38
process-oriented analysis, 52
SADT and, 172ff.
scope of, 7, 10–11, 27ff., 37–38, 40–41, 49
semantics, xix, 4, 55–80, 81, 106, 173
standard for, xix, 9, 12, 55, 107, 133
syntax, xix, 2–4, 50, 81, 106, 173
IDEF0 rules, xix, 106, 107–22
for arrows, 87, 107, 110–15, 204
for boxes, 107–10
for diagrams, 55ff., 107, 115–17
for flow restriction, 114
language, 107ff.
for layout, 87
for models, 63ff., 107, 117–22
summary of, 107
of three to six boxes, 9, 43, 115–17
for working, 23
Input arrow, 11, 56, 86, 133
control vs., 132–33
convention for, 109
Input-control-output-mechanism (ICOM)
codes, 66–70, 107, 112, 116, 210
Integrated Computer-Aided Manufacturing
(ICAM), 5, 6, 10, 13, 16–17, 142, 173
International Society for Enterprise Engineering
(ISEE), 174
International Telephone and Telegraph (ITT),
5, 6, 13, 17
Interview, see Modeling interviews
Jacobson, Ivar, 123
Librarian, 84, 88, 102, 104
FEO diagram of, 97–98
Manufacturing, 7, 10, 24ff., 35ff.
agile, 25, 35
example, 7, 24ff., 118ff.
Manufacturing Technology (ManTech)
Office, 4, 7, 173
Mechanism, 2, 48, 56, 109
arrow, 4, 6, 47, 48, 71–73
arrow branch, 72–73
call arrow vs., 78–79
control vs., 132
levels of, 51–52
multiple, 72
syntax, 3, 56, 67, 71
Model
See also Approval levels; AS-IS model;
Decomposition; Modeling; Quality of
model; Text; TO-BE model; Viewpoint
of model
of activities, 83–84, 86
authoring, 61–62, 82, 83, 84
bad, creation of, 13
boundary, 33
breakdown, 9–10, 81, 82–84
bundling, 82, 86, 98–99
clustering, 82, 86, 142
complexity, 9, 11, 20
for configuration management, 166–71
content-free, 135, 136
of enterprise, 3, 13, 20, 29, 34, 35, 36,
156ff., 160–65
functional, 14
of function vs. process, 147–49, 166
generic, 29, 156–60
glossary, 62–63, 88
level of abstraction, 51–52
library of, 142
migration project, 185, 210
motivational factor, 31–33
presentation of, 82, 97, 105
publication format of, 60, 61, 62–63, 82,
89–91
purpose of, 41, 49, 64, 81, 82–84, 116,
117–18, 123, 128, 135–37, 182ff., 199, 210
purpose statement, 21, 186–87
reuse of, 17, 142
scope of, 7, 10–11, 33, 40–41, 211
strawman, 22
structure of, 31–33, 39, 44, 63–65
types of, 147ff.
validation of, 88–89
Modeling, 12, 17, 23
See also Function modeling; IDEF0
rules; Process modeling; Viewpoint
of model
case study, 179–206
cost of, 141–43
do's and don'ts of, 106–44
of function vs. design, 39–40, 134,
137–40
graphical, vii, 6, 17, 20, 23, 82, 86
intermodel tie syntax for, 92
interviews, 20, 22, 82, 84, 85, 179, 180ff.
management and, 140–43
metrics and, 42
reference language for, 66ff.
reviews, 82, 84
rules, 63ff., 107, 117–22
steps in, 182–205
tools, vii, xx, 13, 28, 34, 105, 204
users and, xix, 4, 6, 22, 23
National Institute of Standards and
Technology (NIST), 9, 12, 14, 174
Node, 65
diagram, 31–33, 65–66, 95–96
number, 60, 64, 65, 66, 104, 107
structure, 65
Office of the Assistant Secretary of
Defense (OASD), 15
Output arrow, 56, 108–9
Parent/child connectivity, 107, 116
Parentheses, arrows and, 112
Parnas design method, 1
Pragmatics, 8–9, 80, 81–105
elements of, 81–82
Process modeling, 10, 19, 24ff., 33ff., 148,
207
Project, 19
budget overruns, 24–25
consensus, 21, 22, 23
deliverables, 23, 35, 36
management, 12ff., 19, 22, 25ff., 37–38,
106, 140–43
personnel, 13, 21, 25–27, 84, 140–41
support for, 25ff., 140–41
Q-label quantification, 173–74
Quality of model, 6, 14, 20–21, 31, 135
arrow bundling and, 129–31
INDEX

arrow content and, 132
arrow labels and, 124–26, 131
box names and, 124–26
controls and, 132–33
diagram simplicity and, 127–29
draft, 14
fog factor and, 124
measurement of, 106, 122
questions for, 123
reader/author cycle and, 122
recommendations for, 135–40
text and, 134
viewpoint and, 133–34
walkthrough session and, 122
Reader, 211
responsibility, 84, 102, 103–5
selection of, 13, 104, 140, 204–5
kit for, 84, 88, 103–5, 203, 204
Reengineering, see Business process reengineering
Reference expression, see Direct reference expression
Request for proposal (RFP), 1, 27–29, 128
Requirements definition, 22
Review
committee, 84, 102, 104, 140
management role in, 102, 104
Ross, Douglas T., v, xvii, 4, 5, 6, 8, 9, 10, 172
Schematic diagram, 82, 95, 96–97, 105
SofTech, Inc., xvii, 5, 172, 173
Software Engineering Institute (SEI) performance assessment, 37
Split-by-type (SBT) structure, 76, 149, 153–55
Staff issues, 4, 12, 13, 16, 22–23, 25ff., 31
Strassmann, Paul, 16, 17–18
Structured Analysis and Design Technique (SADT), xx, 6–7, 57, 172ff.
activity, 83
BPR and, 16, 33
breakdown, 83
data modeling, 54, 173, 174, 175ff.
feedback from analysis effort and, 20
history of, 172–74
IDEF1X and, 54, 175
IDEFO and, v, 13, 54, 61, 172ff.
principles of, 6
Q-label quantification, 173–74
System, 11, 20, 25–27, 211
development, 10, 11, 12, 24ff.
Systems analysis, xix, 11, 13ff., 20ff.
See also AS-IS method; Data analysis;
IDEFO method; Structured Analysis and Design Technique (SADT)
AS-IS model and, 24ff.
Ben’s Burgers case study, 179ff.
CASE and, 12–13
IDEFO method and, 13–14
Text, 20, 23, 61, 64, 89, 134, 211
to-be model, 13, 18, 24, 111, 179, 184ff., 211
in accounting department example, 27
in BPR process, 19, 21, 23
defined, 211
in TQM diagram, 34, 35
Total Quality Management (TQM), 25, 33–36, 176, 186
Traceability, 6
Viewpoint of model, 29–33, 41, 49, 64, 81, 82–84, 92–95, 117ff., 123, 133–34, 166, 182ff.
 decomposition of, 122
defined, 211
field office example and, 29–33
financial, 92, 119, 121
human-factors, 29ff.
management, 92, 93, 94, 119
mixed, 118, 134
motivational factor models, 31ff.
quality, 119, 120, 121, 183
table of, 119
technical, 92, 93, 94, 119, 120, 121
Walkthrough session, 85, 96–97, 105, 122–23, 134
 schematics used in, 96–97, 105
What-if, 71, 129, 203
Workshop approach, 21ff.
users and, 22
Yourdon design method, 1, 12
Zachman, John, 37