

# SOA

# Principles of Service Design

Thomas Erl



PRENTICE HALL

UPPER SADDLE RIVER, NJ • BOSTON • INDIANAPOLIS • SAN FRANCISCO

NEW YORK • TORONTO • MONTREAL • LONDON • MUNICH • PARIS • MADRID

CAPETOWN • SYDNEY • TOKYO • SINGAPORE • MEXICO CITY

# Contents

<b>Preface</b> .....	<b>xxv</b>
<b>Chapter 1: Introduction</b> .....	<b>1</b>
1.1 Objectives of this Book .....	3
1.2 Who this Book Is For .....	3
1.3 What this Book Does Not Cover .....	4
Topics Covered by Other Books .....	4
SOA Standardization Efforts .....	5
1.4 How this Book Is Organized .....	6
Part I: Fundamentals .....	7
Part II: Design Principles .....	9
Part III: Supplemental .....	12
Appendices .....	12
1.5 Symbols, Figures, and Style Conventions .....	13
Symbol Legend .....	13
How Color Is Used .....	13
The Service Symbol .....	13
1.6 Additional Information .....	16
Updates, Errata, and Resources ( <a href="http://www.soabooks.com">www.soabooks.com</a> ) .....	16
Master Glossary ( <a href="http://www.soaglossary.com">www.soaglossary.com</a> ) .....	16
Referenced Specifications ( <a href="http://www.soaspecs.com">www.soaspecs.com</a> ) .....	16
Service-Oriented Computing Poster ( <a href="http://www.soaposters.com">www.soaposters.com</a> ) .....	16

The SOA Magazine ([www.soamag.com](http://www.soamag.com)) . . . . . 17  
 Notification Service . . . . . 17  
 Contact the Author . . . . . 17

**Chapter 2: Case Study . . . . . 19**

2.1 Case Study Background: Cutit Saws Ltd. . . . . 20  
     History . . . . . 20  
     Technical Infrastructure and Automation Environment . . . . . 21  
     Business Goals and Obstacles . . . . . 21

**PART I: FUNDAMENTALS**

**Chapter 3: Service-Oriented Computing and SOA. . . . . 25**

3.1 Design Fundamentals . . . . . 26  
     Design Characteristic . . . . . 27  
     Design Principle . . . . . 28  
     Design Paradigm . . . . . 29  
     Design Pattern . . . . . 30  
     Design Pattern Language . . . . . 31  
     Design Standard . . . . . 32  
     Best Practice . . . . . 34  
     A Fundamental Design Framework . . . . . 35  
 3.2 Introduction to Service-Oriented Computing . . . . . 37  
     Service-Oriented Architecture . . . . . 38  
     Service-Oriented Architecture, Services, and Service-Oriented  
       Solution Logic . . . . . 39  
     Service Compositions . . . . . 39  
     Service Inventory . . . . . 40  
     Understanding Service-Oriented Computing Elements . . . . . 40  
     Service Models . . . . . 43  
     SOA and Web Services . . . . . 46  
     Service Inventory Blueprints . . . . . 51  
     Service-Oriented Analysis and Service Modeling . . . . . 52

- Service-Oriented Design . . . . . 53
- Service-Oriented Architecture: Concepts, Technology,  
and Design . . . . . 54
- 3.3 Goals and Benefits of Service-Oriented Computing . . . . . 55
  - Increased Intrinsic Interoperability . . . . . 56
  - Increased Federation . . . . . 58
  - Increased Vendor Diversification Options . . . . . 59
  - Increased Business and Technology Domain Alignment . . . . . 60
  - Increased ROI . . . . . 61
  - Increased Organizational Agility . . . . . 63
  - Reduced IT Burden . . . . . 64
- 3.4 Case Study Background . . . . . 66

**Chapter 4: Service-Orientation . . . . . 67**

- 4.1 Introduction to Service-Orientation . . . . . 68
  - Services in Business Automation . . . . . 69
  - Services Are Collections of Capabilities . . . . . 69
  - Service-Orientation as a Design Paradigm . . . . . 70
  - Service-Orientation and Interoperability . . . . . 74
- 4.2 Problems Solved by Service-Orientation . . . . . 75
  - Life Before Service-Orientation . . . . . 76
  - The Need for Service-Orientation . . . . . 81
- 4.3 Challenges Introduced by Service-Orientation . . . . . 85
  - Design Complexity . . . . . 85
  - The Need for Design Standards . . . . . 86
  - Top-Down Requirements . . . . . 86
  - Counter-Agile Service Delivery in Support of Agile  
Solution Delivery . . . . . 87
  - Governance Demands . . . . . 88
- 4.4 Additional Considerations . . . . . 89
  - It Is Not a Revolutionary Paradigm . . . . . 89
  - Enterprise-wide Standardization Is Not Required . . . . . 89
  - Reuse Is Not an Absolute Requirement . . . . . 90

- 4.5 Effects of Service-Orientation on the Enterprise . . . . . 91
  - Service-Orientation and the Concept of “Application” . . . . . 91
  - Service-Orientation and the Concept of “Integration”. . . . . 92
  - The Service Composition . . . . . 94
  - Application, Integration, and Enterprise Architectures . . . . . 95
- 4.6 Origins and Influences of Service-Orientation . . . . . 96
  - Object-Orientation. . . . . 97
  - Web Services . . . . . 98
  - Business Process Management (BPM) . . . . . 98
  - Enterprise Application Integration (EAI) . . . . . 98
  - Aspect-Oriented Programming (AOP) . . . . . 99
- 4.7 Case Study Background . . . . . 100

**Chapter 5: Understanding Design Principles . . . . . 103**

- 5.1 Using Design Principles . . . . . 104
  - Incorporate Principles within Service-Oriented Analysis . . . . . 105
  - Incorporate Principles within Formal Design Processes. . . . . 106
  - Establish Supporting Design Standards. . . . . 107
  - Apply Principles to a Feasible Extent . . . . . 108
- 5.2 Principle Profiles . . . . . 109
- 5.3 Design Pattern References . . . . . 111
- 5.4 Principles that Implement vs. Principles that Regulate. . . 111
- 5.5 Principles and Service Implementation Mediums. . . . . 114
  - “Capability” vs. “Operation” vs. “Method” . . . . . 115
- 5.6 Principles and Design Granularity . . . . . 115
  - Service Granularity . . . . . 116
  - Capability Granularity . . . . . 116
  - Data Granularity . . . . . 116
  - Constraint Granularity . . . . . 117
  - Sections on Granularity Levels . . . . . 118
- 5.7 Case Study Background . . . . . 119
  - The Lab Project Business Process . . . . . 119

## **PART II: DESIGN PRINCIPLES**

- Chapter 6: Service Contracts (Standardization and Design) . . . . . 125**
- 6.1 Contracts Explained . . . . . 126
  - Technical Contracts in Abstract . . . . . 126
  - Origins of Service Contracts . . . . . 127
- 6.2 Profiling this Principle . . . . . 130
- 6.3 Types of Service Contract Standardization . . . . . 132
  - Standardization of Functional Service Expression . . . . . 133
  - Standardization of Service Data Representation . . . . . 134
  - Standardization of Service Policies . . . . . 137
- 6.4 Contracts and Service Design . . . . . 140
  - Data Representation Standardization and Transformation Avoidance. . . . . 140
  - Standardization and Granularity . . . . . 142
  - Standardized Service Contracts and Service Models . . . . . 144
  - How Standardized Service Contract Design Affects Other Principles . . . . . 144
- 6.5 Risks Associated with Service Contract Design . . . . . 149
  - Versioning . . . . . 149
  - Technology Dependencies . . . . . 150
  - Development Tool Deficiencies. . . . . 151
- 6.6 More About Service Contracts . . . . . 152
  - Non-Technical Service Contract Documents . . . . . 152
  - “Web Service Contract Design for SOA”. . . . . 153
- 6.7 Case Study Example. . . . . 154
  - Planned Services . . . . . 154
  - Design Standards . . . . . 155
  - Standardized WSDL Definition Profiles . . . . . 155
  - Standardized XML Schema Definitions. . . . . 157
  - Standardized Service and Data Representation Layers . . . . . 157
  - Service Descriptions . . . . . 158
  - Conclusion . . . . . 160

## **Chapter 7: Service Coupling (Intra-Service and Consumer Dependencies) . . . . . 163**

7.1 Coupling Explained . . . . .	164
Coupling in Abstract . . . . .	165
Origins of Software Coupling . . . . .	165
7.2 Profiling this Principle . . . . .	167
7.3 Service Contract Coupling Types . . . . .	169
Logic-to-Contract Coupling (the coupling of service logic to the service contract) . . . . .	173
Contract-to-Logic Coupling (the coupling of the service contract to its logic) . . . . .	174
Contract-to-Technology Coupling (the coupling of the service contract to its underlying technology) . . . . .	176
Contract-to-Implementation Coupling (the coupling of the service contract to its implementation environment) . . . . .	177
Contract-to-Functional Coupling (the coupling of the service contract to external logic) . . . . .	180
7.4 Service Consumer Coupling Types . . . . .	181
Consumer-to-Implementation Coupling . . . . .	182
Standardized Service Coupling and Contract Centralization . . . . .	185
Consumer-to-Contract Coupling . . . . .	185
Measuring Consumer Coupling . . . . .	191
7.5 Service Loose Coupling and Service Design . . . . .	193
Coupling and Service-Orientation . . . . .	193
Service Loose Coupling and Granularity . . . . .	195
Coupling and Service Models . . . . .	196
How Service Loose Coupling Affects Other Principles . . . . .	197
7.6 Risks Associated with Service Loose Coupling . . . . .	200
Limitations of Logic-to-Contract Coupling . . . . .	200
Problems when Schema Coupling Is “too loose” . . . . .	201
7.7 Case Study Example . . . . .	202
Coupling Levels of Existing Services . . . . .	202
Introducing the InvLegacyAPI Service . . . . .	203
Service Design Options . . . . .	205

**Chapter 8: Service Abstraction (Information Hiding and Meta Abstraction Types) . . . . . 211**

- 8.1 Abstraction Explained . . . . . 212
  - Origins of Information Hiding . . . . . 213
- 8.2 Profiling this Principle . . . . . 214
  - Why Service Abstraction Is Needed . . . . . 214
- 8.3 Types of Meta Abstraction . . . . . 218
  - Technology Information Abstraction . . . . . 219
  - Functional Abstraction . . . . . 221
  - Programmatic Logic Abstraction. . . . . 222
  - Quality of Service Abstraction. . . . . 224
  - Meta Abstraction Types and the Web Service Regions of Influence . . . . . 225
  - Meta Abstraction Types in the Real World . . . . . 227
- 8.4 Measuring Service Abstraction. . . . . 231
  - Contract Content Abstraction Levels . . . . . 231
  - Access Control Levels. . . . . 232
  - Abstraction Levels and Quality of Service Meta Information . . . 234
- 8.5 Service Abstraction and Service Design . . . . . 235
  - Service Abstraction vs. Service Encapsulation. . . . . 235
  - How Encapsulation Can Affect Abstraction . . . . . 235
  - Service Abstraction and Non-Technical Contract Documents . . 237
  - Service Abstraction and Granularity . . . . . 238
  - Service Abstraction and Service Models . . . . . 239
  - How Service Abstraction Affects Other Principles . . . . . 239
- 8.6 Risks Associated with Service Abstraction . . . . . 242
  - Multi-Consumer Coupling Requirements . . . . . 242
  - Misjudgment by Humans . . . . . 242
  - Security and Privacy Concerns. . . . . 243
- 8.7 Case Study Example. . . . . 244
  - Service Abstraction Levels . . . . . 244
  - Operation-Level Abstraction Examples . . . . . 247



**Chapter 9: Service Reusability (Commercial and Agnostic Design) . . . . . 253**

- 9.1 Reuse Explained . . . . . 254
  - Reuse in Abstract . . . . . 254
  - Origins of Reuse . . . . . 257
- 9.2 Profiling this Principle . . . . . 259
- 9.3 Measuring Service Reusability and Applying
  - Commercial Design. . . . . 262
  - Commercial Design Considerations . . . . . 262
  - Measures of Planned Reuse . . . . . 265
  - Measuring Actual Reuse . . . . . 267
  - Commercial Design Versus Gold-Plating . . . . . 267
- 9.4 Service Reuse in SOA . . . . . 268
  - Reuse and the Agnostic Service . . . . . 268
  - The Service Inventory Blueprint . . . . . 269
- 9.5 Standardized Service Reuse and Logic Centralization . . 270
  - Understanding Logic Centralization . . . . . 271
  - Logic Centralization as an Enterprise Standard . . . . . 272
  - Logic Centralization and Contract Centralization . . . . . 272
  - Centralization and Web Services . . . . . 274
  - Challenges to Achieving Logic Centralization . . . . . 274
- 9.6 Service Reusability and Service Design . . . . . 276
  - Service Reusability and Service Modeling . . . . . 276
  - Service Reusability and Granularity . . . . . 277
  - Service Reusability and Service Models. . . . . 278
  - How Service Reusability Affects Other Principles . . . . . 278
- 9.7 Risks Associated with Service Reusability and
  - Commercial Design. . . . . 281
  - Cultural Concerns . . . . . 281
  - Governance Concerns . . . . . 283
  - Reliability Concerns . . . . . 286
  - Security Concerns. . . . . 286
  - Commercial Design Requirement Concerns. . . . . 286
  - Agile Delivery Concerns . . . . . 287

- 9.8 Case Study Example . . . . . 288
  - The Inventory Service Profile . . . . . 288
  - Assessing Current Capabilities . . . . . 289
  - Modeling for a Targeted Measure of Reusability . . . . . 289
  - The New EditItemRecord Operation . . . . . 290
  - The New ReportStockLevels Operation . . . . . 290
  - The New AdjustItemsQuantity Operation . . . . . 291
  - Revised Inventory Service Profile . . . . . 292

**Chapter 10: Service Autonomy (Processing Boundaries and Control) . . . . . 293**

- 10.1 Autonomy Explained . . . . . 294
  - Autonomy in Abstract . . . . . 294
  - Origins of Autonomy . . . . . 295
- 10.2 Profiling this Principle . . . . . 296
- 10.3 Types of Service Autonomy . . . . . 297
  - Runtime Autonomy (execution) . . . . . 298
  - Design-Time Autonomy (governance) . . . . . 298
- 10.4 Measuring Service Autonomy . . . . . 300
  - Service Contract Autonomy (services with normalized contracts) . . . . . 301
  - Shared Autonomy . . . . . 305
  - Service Logic Autonomy (partially isolated services) . . . . . 306
  - Pure Autonomy (isolated services) . . . . . 308
  - Services with Mixed Autonomy . . . . . 310
- 10.5 Autonomy and Service Design . . . . . 311
  - Service Autonomy and Service Modeling . . . . . 311
  - Service Autonomy and Granularity . . . . . 311
  - Service Autonomy and Service Models . . . . . 312
  - How Service Autonomy Affects Other Principles . . . . . 314
- 10.6 Risks Associated with Service Autonomy . . . . . 317
  - Misjudging the Service Scope . . . . . 317
  - Wrapper Services and Legacy Logic Encapsulation . . . . . 318
  - Overestimating Service Demand . . . . . 318

- 10.7 Case Study Example . . . . . 319
  - Existing Implementation Autonomy of the GetItem Operation . . 319
  - New Operation-Level Architecture with Increased Autonomy . . 320
  - Effect on the Run Lab Project Composition . . . . . 322

**Chapter 11: Service Statelessness (State Management Deferral and Stateless Design) . . . . . 325**

- 11.1 State Management Explained . . . . . 327
  - State Management in Abstract . . . . . 327
  - Origins of State Management . . . . . 328
  - Deferral vs. Delegation . . . . . 331
- 11.2 Profiling this Principle . . . . . 331
- 11.3 Types of State . . . . . 335
  - Active and Passive . . . . . 335
  - Stateless and Stateful . . . . . 336
  - Session and Context Data . . . . . 336
- 11.4 Measuring Service Statelessness . . . . . 339
  - Non-Deferred State Management (low-to-no statelessness) . . 340
  - Partially Deferred Memory (reduced statefulness) . . . . . 340
  - Partial Architectural State Management Deferral (moderate statelessness) . . . . . 341
  - Full Architectural State Management Deferral (high statelessness) . . . . . 342
  - Internally Deferred State Management (high statelessness) . . 342
- 11.5 Statelessness and Service Design . . . . . 343
  - Messaging as a State Deferral Option . . . . . 343
  - Service Statelessness and Service Instances . . . . . 344
  - Service Statelessness and Granularity . . . . . 346
  - Service Statelessness and Service Models . . . . . 346
  - How Service Statelessness Affects Other Principles . . . . . 347
- 11.6 Risks Associated with Service Statelessness . . . . . 349
  - Dependency on the Architecture . . . . . 349
  - Increased Runtime Performance Demands . . . . . 350
  - Underestimating Delivery Effort . . . . . 350

- 11.7 Case Study Example . . . . . 351
  - Solution Architecture with State Management Deferral . . . . . 352
  - Step 1 . . . . . 353
  - Step 2 . . . . . 354
  - Step 3 . . . . . 355
  - Step 4 . . . . . 356
  - Step 5 . . . . . 357
  - Step 6 . . . . . 358
  - Step 7 . . . . . 359

**Chapter 12: Service Discoverability (Interpretability and Communication) . . . . . 361**

- 12.1 Discoverability Explained . . . . . 362
  - Discovery and Interpretation, Discoverability and Interpretability in Abstract . . . . . 364
  - Origins of Discovery . . . . . 367
- 12.2 Profiling this Principle . . . . . 368
- 12.3 Types of Discovery and Discoverability
  - Meta Information . . . . . 371
  - Design-Time and Runtime Discovery . . . . . 371
  - Discoverability Meta Information . . . . . 373
  - Functional Meta Data . . . . . 374
  - Quality of Service Meta Data . . . . . 374
- 12.4 Measuring Service Discoverability . . . . . 375
  - Fundamental Levels . . . . . 375
  - Custom Rating System . . . . . 376
- 12.5 Discoverability and Service Design . . . . . 376
  - Service Discoverability and Service Modeling . . . . . 377
  - Service Discoverability and Granularity . . . . . 378
  - Service Discoverability and Policy Assertions . . . . . 378
  - Service Discoverability and Service Models . . . . . 378
  - How Service Discoverability Affects Other Principles . . . . . 378

- 12.6 Risks Associated with Service Discoverability . . . . . 381
  - Post-Implementation Application of Discoverability . . . . . 381
  - Application of this Principle by Non-Communicative Resources 381
- 12.7 Case Study Example . . . . . 382
  - Service Profiles (Functional Meta Information) . . . . . 382
  - Related Quality of Service Meta Information . . . . . 386

**Chapter 13: Service Composability (Composition Member Design and Complex Compositions) . . . . . 387**

- 13.1 Composition Explained . . . . . 388
  - Composition in Abstract . . . . . 388
  - Origins of Composition . . . . . 390
- 13.2 Profiling this Principle . . . . . 392
- 13.3 Composition Concepts and Terminology . . . . . 396
  - Compositions and Composition Instances . . . . . 397
  - Composition Members and Controllers . . . . . 398
  - Service Compositions and Web Services . . . . . 401
  - Service Activities . . . . . 402
  - Composition Initiators . . . . . 403
  - Point-to-Point Data Exchanges and Compositions . . . . . 405
  - Types of Compositions . . . . . 406
- 13.4 The Complex Service Composition . . . . . 407
  - Stages in the Evolution of a Service Inventory . . . . . 407
  - Defining the Complex Service Composition . . . . . 410
  - Preparing for the Complex Service Composition . . . . . 411
- 13.5 Measuring Service Composability and Composition
  - Effectiveness Potential . . . . . 412
  - Evolutionary Cycle States of a Composition . . . . . 412
  - Composition Design Assessment . . . . . 413
  - Composition Runtime Assessment . . . . . 415
  - Composition Governance Assessment . . . . . 417
  - Measuring Composability . . . . . 419

- 13.6 Composition and Service Design . . . . . 427
  - Service Composability and Granularity. . . . . 427
  - Service Composability and Service Models . . . . . 428
  - Service Composability and Composition Autonomy. . . . . 430
  - Service Composability and Orchestration. . . . . 430
  - How Service Composability Affects Other Principles . . . . . 432
- 13.7 Risks Associated with Service Composition . . . . . 437
  - Composition Members as Single Points of Failure . . . . . 437
  - Composition Members as Performance Bottlenecks . . . . . 437
  - Governance Rigidity of “Over-Reuse” in Compositions . . . . . 438
- 13.8 Case Study Example. . . . . 439

**PART III: SUPPLEMENTAL**

**Chapter 14: Service-Oriented and Object-Oriented: A Comparison of Principles and Concepts . . . . . 445**

- 14.1 A Tale of Two Design Paradigms . . . . . 446
- 14.2 A Comparison of Goals. . . . . 449
  - Increased Business Requirements Fulfillment . . . . . 450
  - Increased Robustness . . . . . 451
  - Increased Extensibility . . . . . 451
  - Increased Flexibility. . . . . 452
  - Increased Reusability and Productivity. . . . . 452
- 14.3 A Comparison of Fundamental Concepts. . . . . 453
  - Classes and Objects. . . . . 453
  - Methods and Attributes. . . . . 454
  - Messages . . . . . 454
  - Interfaces . . . . . 456
- 14.4 A Comparison of Design Principles . . . . . 457
  - Encapsulation . . . . . 458
  - Inheritance . . . . . 459

- Generalization and Specialization . . . . . 461
- Abstraction . . . . . 463
- Polymorphism . . . . . 463
- Open-Closed Principle (OCP) . . . . . 465
- Don't Repeat Yourself (DRY) . . . . . 465
- Single Responsibility Principle (SRP) . . . . . 466
- Delegation . . . . . 468
- Association . . . . . 469
- Composition . . . . . 470
- Aggregation . . . . . 471
- 14.5 Guidelines for Designing Service-Oriented Classes. . . . . 472
  - Implement Class Interfaces . . . . . 473
  - Limit Class Access to Interfaces . . . . . 473
  - Do Not Define Public Attributes in Interfaces . . . . . 473
  - Use Inheritance with Care . . . . . 473
  - Avoid Cross-Service “has-a” Relationships . . . . . 474
  - Use Abstract Classes for Modeling, Not Design . . . . . 474
  - Use Façade Classes . . . . . 474

**Chapter 15: Supporting Practices . . . . . 477**

- 15.1 Service Profiles . . . . . 478
  - Service-Level Profile Structure . . . . . 478
  - Capability Profile Structure . . . . . 480
  - Additional Considerations . . . . . 482
- 15.2 Vocabularies . . . . . 483
  - Service-Oriented Computing Terms . . . . . 484
  - Service Classification Terms . . . . . 484
  - Types and Associated Terms . . . . . 485
  - Design Principle Application Levels . . . . . 487
- 15.3 Organizational Roles . . . . . 488
  - Service Analyst . . . . . 490
  - Service Architect . . . . . 490
  - Service Custodian . . . . . 491
  - Schema Custodian . . . . . 491
  - Policy Custodian . . . . . 492

Service Registry Custodian . . . . . 492  
 Technical Communications Specialist . . . . . 493  
 Enterprise Architect . . . . . 493  
 Enterprise Design Standards Custodian (and Auditor) . . . . . 494

**Chapter 16: Mapping Service-Orientation Principles to Strategic Goals . . . . . 497**

16.1 Principles that Increase Intrinsic Interoperability . . . . . 498  
 16.2 Principles that Increase Federation . . . . . 501  
 16.3 Principles that Increase Vendor Diversification Options . 501  
 16.4 Principles that Increase Business and Technology Domain Alignment . . . . . 502  
 16.5 Principles that Increase ROI . . . . . 504  
 16.6 Principles that Increase Organizational Agility . . . . . 505  
 16.7 Principles that Reduce the Overall Burden of IT . . . . . 507

**PART IV: APPENDICES**

**Appendix A: Case Study Conclusion . . . . . 513**

**Appendix B: Process Descriptions . . . . . 517**

B.1 Delivery Processes . . . . . 518  
 Bottom-Up vs. Top-Down . . . . . 518  
 The Inventory Analysis Cycle . . . . . 520  
 Inventory Analysis and Service-Oriented Design . . . . . 521  
 Choosing a Delivery Strategy . . . . . 521  
 B.2 Service-Oriented Analysis Process . . . . . 522  
 Define Analysis Scope . . . . . 522  
 Identify Affected Systems . . . . . 523  
 Perform Service Modeling . . . . . 523



B.3 Service Modeling Process . . . . . 523

B.4 Service-Oriented Design Processes. . . . . 525

    Design Processes and Service Models . . . . . 526

    Service Design Processes and Service-Orientation . . . . . 527

**Appendix C: Principles and Patterns**

**Cross-Reference . . . . . 529**

**Additional Resources . . . . . 533**

**About the Author . . . . . 535**

**About the Photos . . . . . 537**

**Index . . . . . 539**