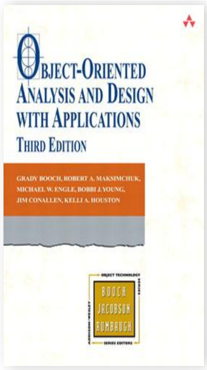
Sign In START FREE TRIAL


OBJECT-ORIENTED ANALYSIS AND DESIGN WITH APPLICATIONS
THIRD EDITION
GRADY BOOCH, ROBERT A. MAKSIMCHUK, MICHAEL W. ENGLE, BOBBI J. YOUNG, JIM CONALLEN, KELLI A. HOUSTON
BOOCH JACOBSON RUMBAUGH

Object-Oriented Analysis and Design with Applications, Third Edition

6 reviews

by Bobbi J. Young Ph.D., Kelli A. Houston, Jim Conallen, Michael W. Engle, Robert A. Maksimchuk, Grady Booch

Publisher: Addison-Wesley Professional

Release Date: April 2007

ISBN: 9780201895513

Topics: Software Development

View table of contents

START READING

Book Description

Object-Oriented Design with Applications has long been the essential reference to object-oriented technology, which, in turn, has evolved to join the mainstream of industrial-strength software development. In this third edition--the first revision in 13 years--readers can learn to apply object-oriented methods using new paradigms such as Java, the Unified Modeling Language (UML) 2.0, and .NET.

The authors draw upon their rich and varied experience to offer improved methods for object development and numerous examples that tackle the complex problems faced by software engineers, including systems architecture, data acquisition, cryptoanalysis, control systems, and Web development. They illustrate essential concepts, explain the method, and show successful applications in a variety of fields. You'll also find pragmatic advice on a host of issues, including classification, implementation strategies, and cost-effective project management.

New to this new edition are

-

- New domains and contexts
- A greatly enhanced focus on modeling--as eagerly requested by readers--with five chapters that each delve into one phase of the overall development lifecycle.
- Fresh approaches to reasoning about complex systems
- An examination of the conceptual foundation of the widely misunderstood fundamental elements of the object model, such as abstraction, encapsulation, modularity, and hierarchy
- How to allocate the resources of a team of developers and mange the risks associated with developing complex software systems
- An appendix on object-oriented programming languages

This is the seminal text for anyone who wishes to use object-oriented technology to manage the complexity inherent in many kinds of systems.

Sidebars
Preface
Acknowledgments
About the Authors
Section I: Concepts
Chapter 1: Complexity
Chapter 2: The Object Model
Chapter 3: Classes and Objects
Chapter 4: Classification
Section II: Method
Chapter 5: Notation
Chapter 6: Process
Chapter 7: Pragmatics
Chapter 8: System Architecture: Satellite-Based Navigation

Sign In START FREE TRIAL

Chapter 12: Web Application: Vacation Tracking System
Appendix A: Object-Oriented Programming Languages
Appendix B: Further Reading
Notes
Glossary
Classified Bibliography
Index

Table of Contents

Copyright
- Dedication

The Addison-Wesley Object Technology Series
- The Component Software Series

Sidebars

Preface
- Changes to the Second Edition
- Goals
- Audience
- Structure
  - Concepts
  - Method
  - Applications
  - Supplemental Material
  - A Note about Tools
- Using This Book

Acknowledgments

About the Authors

I. Concepts

Sign In START FREE TRIAL

The Structure of a Personal Computer

The Structure of Plants and Animals

The Structure of Matter

The Structure of Social Institutions

1.2. The Inherent Complexity of Software

Defining Software Complexity

Why Software Is Inherently Complex

The Complexity of the Problem Domain

The Difficulty of Managing the Development Process

The Flexibility Possible through Software

The Problems of Characterizing the Behavior of Discrete Systems

1.3. The Five Attributes of a Complex System

Hierarchic Structure

Relative Primitives

Separation of Concerns

Common Patterns

Stable Intermediate Forms

1.4. Organized and Disorganized Complexity

The Canonical Form of a Complex System

The Limitations of the Human Capacity for Dealing with Complexity

1.5. Bringing Order to Chaos

The Role of Decomposition

Algorithmic Decomposition

Object-Oriented Decomposition

Algorithmic versus Object-Oriented Decomposition

The Role of Abstraction

The Role of Hierarchy

1.6. On Designing Complex Systems

Engineering as a Science and an Art

The Meaning of Design

Sign In START FREE TRIAL

The Models of Object-Oriented Development

Summary

2. The Object Model

2.1. The Evolution of the Object Model

The Generations of Programming Languages

The Topology of First- and Early Second-Generation Programming Languages

The Topology of Late Second- and Early Third-Generation Programming Languages

The Topology of Late Third-Generation Programming Languages

The Topology of Object-Based and Object-Oriented Programming Languages

2.2. Foundations of the Object Model

Object-Oriented Programming

Object-Oriented Design

Object-Oriented Analysis

2.3. Elements of the Object Model

The Meaning of Abstraction

Examples of Abstraction

The Meaning of Encapsulation

Examples of Encapsulation

The Meaning of Modularity

Examples of Modularity

The Meaning of Hierarchy

Examples of Hierarchy: Single Inheritance

Examples of Hierarchy: Multiple Inheritance

Examples of Hierarchy: Aggregation

The Meaning of Typing

Examples of Typing: Static and Dynamic Typing

The Meaning of Concurrency

Examples of Concurrency

The Meaning of Persistence

2.4. Applying the Object Model

Sign In START FREE TRIAL

Summary

3. Classes and Objects

3.1. The Nature of an Object

What Is and What Isn’t an Object

State

Behavior

Operations

Roles and Responsibilities

Objects as Machines

Identity

3.2. Relationships among Objects

Links

Visibility

Synchronization

Aggregation

3.3. The Nature of a Class

What Is and What Isn’t a Class

Interface and Implementation

Class Lifecycle

3.4. Relationships among Classes

Association

Semantic Dependencies

Multiplicity

Inheritance

Single Inheritance

Polymorphism

Multiple Inheritance

Aggregation

Physical Containment

Dependencies

Sign In START FREE TRIAL

The Role of Classes and Objects in Analysis and Design

3.6. On Building Quality Classes and Objects

Measuring the Quality of an Abstraction

Choosing Operations

Functional Semantics

Time and Space Semantics

Choosing Relationships

The Law of Demeter

Mechanisms and Visibility

Choosing Implementations

Representation

Packaging

Summary

4. Classification

4.1. The Importance of Proper Classification

The Difficulty of Classification

The Incremental and Iterative Nature of Classification

4.2. Identifying Classes and Objects

Classical and Modern Approaches

Classical Categorization

Conceptual Clustering

Prototype Theory

Applying Classical and Modern Theories

Object-Oriented Analysis

Classical Approaches

Behavior Analysis

Domain Analysis

Use Case Analysis

CRC Cards

Informal English Description

Sign In START FREE TRIAL

Identifying Key Abstractions

Refining Key Abstractions

Naming Key Abstractions

Identifying Mechanisms

Mechanisms as Patterns

Examples of Mechanisms

Summary

II. Method

5. Notation

5.1. The Unified Modeling Language

A Brief Historical Perspective

Models and Multiple Views

Diagram Taxonomy

Structure Diagrams

Behavior Diagrams

The Use of Diagrams in Practice

Conceptual, Logical, and Physical Models

The Role of Tools

The Products of Object-Oriented Development

Scaling Up and Scaling Down

The Syntax and Semantics of the UML

UML 2.0 Information Sources

5.2. Package Diagrams

Essentials: The Package Notation

Essentials: Visibility of Elements

Essentials: The Dependency Relationship

Essentials: Package Diagrams

Advanced Concepts: Import and Access

5.3. Component Diagrams

Essentials: The Component Notation

Sign In START FREE TRIAL

Essentials: Component Realizations

Advanced Concepts: A Component’s Internal Structure

5.4. Deployment Diagrams

Essentials: The Artifact Notation

Essentials: The Node Notation

Essentials: The Deployment Diagram

5.5. Use Case Diagrams

Essentials: Actors

Essentials: Use Cases

Essentials: The Use Case Diagram

Specifying Use Case Details

An Example Use Case Specification

Use Case Specification

Advanced Concepts: «include» and «extend» Relationships

«include» Relationships

«extend» Relationships

The Dangers of «include» and «extend» Relationships

Advanced Concepts: Generalization

5.6. Activity Diagrams

Essentials: Actions

Essentials: Starting and Stopping

Essentials: Decision and Merge Nodes

Essentials: Partitions

Advanced Concepts: Forks, Joins, and Concurrency

Advanced Concepts: Object Flows

Advanced Concepts: Additional Elements

5.7. Class Diagrams

Essentials: The Class Notation

Essentials: Class Relationships

Advanced Concepts: Template (Parameterized) Classes

Sign In START FREE TRIAL

Advanced Concepts: Constraints

Advanced Concepts: Association Classes and Notes

5.8. Sequence Diagrams

Essentials: Objects and Interactions

Essentials: Lifelines and Messages

Advanced Concepts: Destruction Events

Advanced Concepts: Execution Specification

Advanced Concepts: Interaction Use

Advanced Concepts: Control Constructs

5.9. Interaction Overview Diagrams

Essentials: Frames

Essentials: Flow of Control Elements

Essentials: Interaction Diagram Elements

5.10. Composite Structure Diagrams

Essentials: Composite Structure Parts

Essentials: Composite Structure Ports and Interfaces

Essentials: Composite Structure Connectors

Advanced Concepts: Collaborations

5.11. State Machine Diagrams

Essentials: Initial, Final, and Simple States

Essentials: Transitions and Events

Advanced Concepts: State Activities—Entry, Do, and Exit Activities

Advanced Concepts: Controlling Transitions

Advanced Concepts: Composite States and Nested States

Advanced Concepts: Concurrency and Control

Advanced Concepts: Submachine State

Advanced Concepts: Additional State Machine Diagram Elements

5.12. Timing Diagrams

Essentials: More of the Same

Essentials: Layout

Sign In START FREE TRIAL

Advanced Concepts: Alternate Representations

Advanced Concepts: Events versus Messages

5.13. Object Diagrams

Essentials: Objects

Essentials: Object Relationships

Advanced Concepts: End Names and Qualifiers

5.14. Communication Diagrams

Essentials: Objects, Links, and Messages

Essentials: Sequence Expressions

Advanced Concepts: Messages and Synchronization

Advanced Concepts: Iteration Clauses and Guards

Summary

6. Process

6.1. First Principles

Traits of Successful Projects

Strong Architectural Vision

Iterative and Incremental Lifecycle

Toward a Rational Development Process

6.2. The Macro Process: The Software Development Lifecycle

Overview

The Macro Process Content Dimension—Disciplines

The Macro Process Time Dimension—Milestones and Phases

Inception

Purpose

Activities

Work Products

Milestone: Scope Is Understood

Elaboration

Purpose

Activities

Sign In START FREE TRIAL

Construction

Purpose

Activities

Work Products

Milestone: System Is Ready for End-User Testing

Transition

Purpose

Activities

Work Products

Milestone: System Is Ready to Be Deployed

The Macro Process Time Dimension—Iterations

Release Planning

6.3. The Micro Process: The Analysis and Design Process

Overview

Levels of Abstraction

Activities

Products

The Micro Process and Levels of Abstraction

Identifying Elements

Products

Steps

Milestones and Measures

Defining Element Collaborations

Products

Steps

Behavior Analysis

Pattern Scavenging

Milestones and Measures

Defining Element Relationships

Products

Sign In START FREE TRIAL

Detailing Element Semantics

Products

Steps

Milestones and Measures

Summary

7. Pragmatics

7.1. Management and Planning

Risk Management

Task Planning

Development Reviews

7.2. Staffing

Resource Allocation

Development Team Roles

7.3. Release Management

Configuration Management and Version Control

Integration

Testing

7.4. Reuse

Elements of Reuse

Institutionalizing Reuse

7.5. Quality Assurance and Metrics

Software Quality

Object-Oriented Metrics

7.6. Documentation

Development Legacy

Documentation Contents

7.7. Tools

Kinds of Tools

Organizational Implications

7.8. Special Topics

Sign In START FREE TRIAL

7.9. The Benefits and Risks of Object-Oriented Development

The Benefits of Object-Oriented Development

The Risks of Object-Oriented Development

Summary

III. Applications

8. System Architecture: Satellite-Based Navigation

8.1. Inception

Requirements for the Satellite Navigation System

Defining the Boundaries of the Problem

Determining Mission Use Cases

Determining System Use Cases

8.2. Elaboration

Developing a Good Architecture

Defining Architectural Development Activities

Validating the Proposed System Architecture

Allocating Nonfunctional Requirements and Specifying Interfaces

Stipulating the System Architecture and Its Deployment

Decomposing the System Architecture

8.3. Construction

8.4. Post-Transition

Adding New Functionality

Changing the Target Hardware

9. Control System: Traffic Management

9.1. Inception

Requirements for the Train Traffic Management System

Determining System Use Cases

9.2. Elaboration

Analyzing System Functionality

Defining the TTMS Architecture

From Systems Engineering to Hardware and Software Engineering

Sign In START FREE TRIAL

Message Passing

Train Schedule Planning

Displaying Information

Sensor Data Acquisition

Release Management

System Architecture

Subsystem Specification

9.4. Post-Transition

10. Artificial Intelligence: Cryptanalysis

10.1. Inception

Cryptanalysis Requirements

Defining the Boundaries of the Problem

The Architecture of the Blackboard Framework

Analysis of Knowledge Sources

10.2. Elaboration

Blackboard Objects

Dependencies and Affirmations

10.3. Construction

Designing the Blackboard Objects

Designing the Knowledge Sources

Designing Specialized Knowledge Sources

Generalizing the Knowledge Sources

Designing the Controller

Integrating the Blackboard Framework

Integrating the Topmost Objects

Implementing the Assumption Mechanism

Adding New Knowledge Sources

10.4. Post-Transition

System Enhancements

Changing the Requirements

Requirements for the Weather Monitoring Station

Defining the Boundaries of the Problem

Scenarios

11.2. Elaboration

Weather Monitoring System Use Cases

The Architecture Framework

11.3. Construction

The Frame Mechanism

Release Planning

The Sensor Mechanism

The Display Mechanism

The User Interface Mechanism

11.4. Post-Transition

12. Web Application: Vacation Tracking System

12.1. Inception

The Requirements

The Use Case Model

12.2. Elaboration

The Deployment View

The Logical View

The Process View

The Implementation View

The Use Case View

12.3. Construction

The User Experience Model

The Analysis and Design Models

Entities

Service Data Objects

Primary Key Generation

Finders

Sign In START FREE TRIAL

Populating Dynamic Content

Invoking Business Logic

12.4. Transition and Post-Transition

A. Object-Oriented Programming Languages

A.1. Language Evolution

A.2. Smalltalk

Overview

Example

References

A.3. C++

Overview

Example

References

A.4. Java

Overview

Example

References

B. Further Reading

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Chapter 12

Sign In START FREE TRIAL

Section I: Concepts

Chapter 1: Complexity

Chapter 2: The Object Model

Chapter 3: Classes and Objects

Chapter 4: Classification

Section II: Method

Chapter 5: Notation

Chapter 6: Process

Chapter 7: Pragmatics

Section III: Applications

Chapter 8: System Architecture: Satellite-Based Navigation

Chapter 9: Control System: Traffic Management

Chapter 10: Artificial Intelligence: Cryptanalysis

Chapter 12: Web Application: Vacation Tracking System

Appendix A: Object-Oriented Programming Languages

Glossary

Classified Bibliography

A. Classification

B. Object-Oriented Analysis

C. Object-Oriented Applications

D. Object-Oriented Architectures

E. Object-Oriented Databases

F. Object-Oriented Design

G. Object-Oriented Programming

H. Software Engineering

I. Special References

J. Theory

K. Tools and Environments

L. The Unified Modeling Language

Sign In START FREE TRIAL

Explore

Tour

Pricing

Enterprise

Government

Education

Queue App

Learn

Blog

Contact

Careers

Press Resources

Support

Twitter

GitHub

Facebook

LinkedIn

Terms of Service