

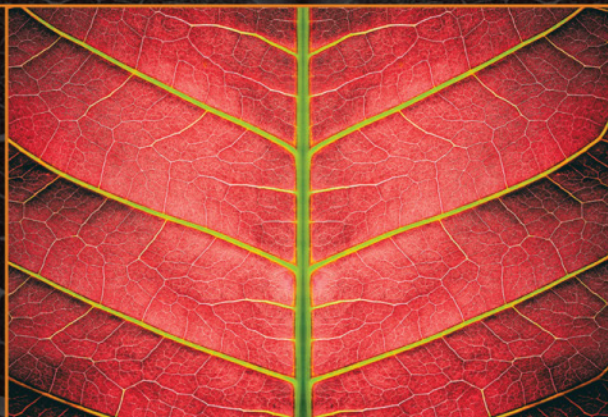
*The Addison Wesley Signature Series*



# BALANCING COUPLING IN SOFTWARE DESIGN

UNIVERSAL DESIGN PRINCIPLES  
FOR ARCHITECTING MODULAR  
SOFTWARE SYSTEMS

VLAD KHONONOV



Forewords by  
REBECCA WIRFS-BROCK  
and KENT BECK

## Praise for *Balancing Coupling in Software Design*

“Your software can get easier to change over time, but it’s hard work to make that happen. With the concepts and skills you’ll gain from this book, though, you will be well on your way.”

—Kent Beck, *Creator of Extreme Programming and Test-Driven Development*

“Coupling is one of those words that is used a lot, but little understood. Vlad propels us from simplistic slogans like ‘always decouple components’ to a nuanced discussion of coupling in the context of complexity and software evolution. If you build modern software, read this book!”

—Gregor Hohpe, *author of The Software Architect Elevator*

“Get ready to unravel the multi-dimensional nature of coupling and the forces at work behind the scenes. The reference for those looking for a means to both assess and understand the real impact of design decisions.”

—Chris Bradford, *Director of Digital Services, Cambridge Consultants*

“Coupling is a tale as old as software. It’s a difficult concept to grasp and explain, but Vlad effortlessly lays out the many facets of coupling in this book, presenting a tangible model to measure and balance coupling in modern distributed systems. This is a must-read for every software professional!”

—Laila Bougria, *solutions architect & engineer*

“This book is essential for every software architect and developer, offering an unparalleled, thorough, and directly applicable exploration of the concept of coupling. Vlad’s work is a crucial resource that will be heavily quoted and referenced in future discussions and publications.”

—Michael Plöd, *fellow @ INNOQ*

“Every software engineer is sensitive to coupling, the measure of interconnection between parts. Still, many times the understanding of such a fundamental property remains unarticulated. In this book, Vlad introduces a much-needed intellectual tool to reason about coupling in a systematic way, offering a novel perspective on this essential topic.”

—Ilio Catallo, *senior software engineer*

“Coupling is among the most slippery topics in software development. However, with this book, Vlad simplifies for us how coupling, from a great villain, can become a design tool when well understood. This is an indispensable guide for anyone dealing with software design—especially complex ones.”

—William Santos, *software architect*

“*Balancing Coupling in Software Design* by Vlad Khononov is an essential read for architects aiming for quality, evolvable systems. Khononov expertly classifies dependencies and reveals how varying designs impact effort based on component distance and change frequency, introducing a unified metric for coupling. With insightful case studies, he guides readers toward achieving optimal modularity and long-term system adaptability by illustrating and rectifying imbalances.”

—Asher Sterkin, *independent software technology expert*

“Khononov’s groundbreaking work unifies paramount forces of software design into a coherent model for evaluating coupling of software systems. His insights provide an invaluable framework for architects to design modular, evolving systems that span legacy and modern architectures.”

—Felipe Henrique Gross Windmoller, *staff software engineer, Banco do Brasil*

“This book systematizes over five decades of software design knowledge, offering a comprehensive guide on coupling, its dimensions, and how to manage it effectively. If software design is a constant battle with complexity, then this book is about mastering the art of winning.”

—Ivan Zakervsky, *IT architect*

# **Balancing Coupling in Software Design: Universal Design Principles for Architecting Modular Software Systems**

## **Table of Contents**

Cover

Title Page

Copyright Page

Contents

Series Editor Foreword

Foreword

Foreword

Preface

Acknowledgments

About the Author

Introduction

Part I: Coupling

Chapter 1: Coupling and System Design

What Is Coupling?

Magnitude of Coupling

Shared Lifecycle

Shared Knowledge

Flow of Knowledge

Systems

Coupling in Systems

Optional: Coupling and Cost Management in Mechanical Engineering



Pearson

# Table of Contents

Key Takeaways

Quiz

## Chapter 2: Coupling and Complexity: Cynefin

What Is Complexity?

Complexity in Software Design

Complexity Is Subjective

Cynefin

Clear

Complicated

Complex

Chaotic

Disorder

Comparing Cynefin Domains

Cynefin in Software Design

Example A: Integrating an External Service

Example B: Changing Database Indexes

Cynefin Applications

Cynefin and Complexity

Key Takeaways

Quiz

## Chapter 3: Coupling and Complexity: Interactions

Nature of Complexity

Complexity and System Design

Linear Interactions

Complex Interactions

Complexity and System Size

Hierarchical Complexity

Optimizing Only the Global Complexity

Optimizing Only the Local Complexity

Balancing Complexity

Degrees of Freedom

# Table of Contents

Degrees of Freedom in Software Design

Degrees of Freedom and Complex Interactions

## Complexity and Constraints

Example: Constraining Degrees of Freedom

Constraints in Cynefin Domains

## Coupling and Complex Interactions

### Example: Connecting Coupling and Complexity

Design A: Using SQL to Filter Support Cases

Design B: Using a Query Object

Design C: Using Specialized Finder Methods

Coupling, Degrees of Freedom, and Constraints

## Key Takeaways

## Quiz

## Chapter 4: Coupling and Modularity

### Modularity

### Modules

LEGO Bricks

Camera Lenses

### Modularity in Software Systems

Software Modules

Function, Logic, and Context of Software Modules

Effective Modules

Modules as Abstractions

### Modularity, Complexity, and Coupling

Deep Modules

Modularity Versus Complexity

Modularity: Too Much of a Good Thing

### Coupling in Modularity

## Key Takeaways

## Quiz

## Part II: Dimensions

# **Table of Contents**

## **Chapter 5: Structured Designs Module Coupling**

Structured Design

Module Coupling

Content Coupling

Common Coupling

External Coupling

Control Coupling

Stamp Coupling

Data Coupling

Comparison of Module Coupling Levels

Key Takeaways

Quiz

## **Chapter 6: Connascence**

What Is Connascence?

Static Connascence

Connascence of Name

Connascence of Type

Connascence of Meaning

Connascence of Algorithm

Connascence of Position

Dynamic Connascence

Connascence of Execution

Connascence of Timing

Connascence of Value

Connascence of Identity

Evaluating Connascence

Managing Connascence

Connascence and Structured Designs Module Coupling

Key Takeaways

Quiz

## **Chapter 7: Integration Strength**

# Table of Contents

## Strength of Coupling

Structured Design, Connascence, or Both?

Structured Design and Connascence: Blind Spots

Different Strategy

## Integration Strength

Running Example: Sharing a Database

## Intrusive Coupling

Examples of Intrusive Coupling

Running Example: Intrusive Coupling by Sharing a Database

Effects of Intrusive Coupling

## Functional Coupling

Degrees of Functional Coupling

Causes for Functional Coupling

Running Example: Functional Coupling by Sharing a Database

Effects of Functional Coupling

## Model Coupling

Degrees of Model Coupling

Running Example: Model Coupling by Sharing a Database

Effects of Model Coupling

## Contract Coupling

Example of Contract Coupling

Degrees of Contract Coupling

Depth of Contract Coupling

Running Example: Contract Coupling by Sharing a Database

Effects of Contract Coupling

## Integration Strength Discussion

Example: Distributed System

## Integration Strength and Asynchronous Execution

## Key Takeaways

## Quiz

## Chapter 8: Distance

Distance and Encapsulation Boundaries



# Table of Contents

Cost of Distance

Distance as Lifecycle Coupling

Evaluating Distance

## Additional Factors Affecting Distance

Distance and Socio-Technical Design

Distance and Runtime Coupling

Asynchronous Communication and Cost of Change

## Distance Versus Proximity

## Distance Versus Integration Strength

## Key Takeaways

## Quiz

## Chapter 9: Volatility

### Changes and Coupling

### Why Software Changes

Solution Changes

Problem Changes

### Evaluating Rates of Changes

Domain Analysis

Source Control Analysis

### Volatility and Integration Strength

### Inferred Volatility

### Key Takeaways

### Quiz

## Part III: Balance

## Chapter 10: Balancing Coupling

### Combining the Dimensions of Coupling

Measurement Units

Stability: Volatility and Strength

Actual Costs: Volatility and Distance

Modularity and Complexity: Strength and Distance

### Combining Strength, Distance, and Volatility



# Table of Contents

Maintenance Effort: Strength, Distance, Volatility

Balanced Coupling: Strength, Distance, Volatility

## Balancing Coupling on a Numeric Scale

Scale

Balanced Coupling Equation

Balanced Coupling: Examples

## Key Takeaways

## Quiz

## Chapter 11: Rebalancing Coupling

### Resilient Design

### Software Change Vectors

Tactical Changes

Strategic Changes

### Rebalancing Coupling

Strength

Volatility

Distance

Rebalancing Complexity

## Key Takeaways

## Quiz

## Chapter 12: Fractal Geometry of Software Design

### Growth

Network-Based Systems

Software Design as a Network-Based System

Why Do Systems Grow?

Growth Limits

Growth Dynamics in Software Design

### Innovation

Innovation in Software Design

Abstraction as Innovation

### Fractal Geometry

# Table of Contents

Fractal Modularity

Key Takeaways

Quiz

## Chapter 13: Balanced Coupling in Practice

Microservices

Case Study 1: Events Sharing Extraneous Knowledge

Case Study 2: Good Enough Integration

Architectural Patterns

Case Study 3: Reducing Complexity

Case Study 4: Layers, Ports, and Adapters

Business Objects

Case Study 5: Entities and Aggregates

Case Study 6: Organizing Classes

Methods

Case Study 7: Divide and Conquer

Case Study 8: Code Smells

Key Takeaways

Quiz

## Chapter 14: Conclusion

Epilogue

Appendix A: The Ballad of Coupling

Appendix B: Glossary of Coupling

Appendix C: Answers to Quiz Questions

Bibliography

Index