

<<UML和模式应用>>

图书基本信息

书名：<<UML和模式应用>>

13位ISBN编号：9787508322049

10位ISBN编号：7508322045

出版时间：2004-5

出版时间：中国电力出版社

作者：（美）Carig Larman,（美）Philippe Kruchten

页数：627

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<UML和模式应用>>

内容概要

本书是当今最畅销的介绍面向对象的分析/设计、迭代开发和UML的书籍。本书的目的是帮助开发人员或学生掌握关于面向对象的分析和设计的核心原理和最佳实践，并将它们真正应用到软件设计中去。本书的作者是颇有声望的对象技术和迭代方法的领军人物。本书在第一版的基础上做了全面的修订，增加了新的案例研究，更新了关于模式、用况、统一过程、架构分析等的探讨。

<<UML和模式应用>>

作者简介

Craig

Larman是Valtech公司的首席科学家，该公司在技术咨询领域居领先地位；在国际软件开发社区中，他是对象技术、模式、UML、建模和迭代开发方面广为人知的专家。

Larman从20世纪80年代就开始帮助人们学习对象和迭代开发，他亲自指导过的开发人员已达数千人此外，他还是《Java 2 Performance and Idiom

Guide》一书的作者之一，并拥有计算机科学学士和硕士学位、Larman的最新作品《Agile&Iterative Development : a Manager's Guide》(《敏捷迭代开发》)即将由中国电力出版社出版。

<<UML和模式应用>>

书籍目录

foreword xv

preface xvii

Part i introduction

1 object-oriented analysis and design

applying uml and patterns in ooa/d

assigning responsibilities

what is analysis and design?

what is object-oriented analysis and design?

an example

the uml

further readings

2 iterative development and the unified process

the most important up idea: iterative development

additional up best practices and concepts

the up phases and schedule-oriented terms

the up disciplines (was workfiows)

process customization and the development case

the agile up

the sequential "waterfall" lifecycle

you know you didn't understand the up when...

further readings

3 case study: the nextgen pos system

the nextgen pos system

architectural layers and case study emphasis

the book's strategy: iterative learning and development

Part ii inception

4 inception

inception: an analogy

inception may be very brief

what artifacts may start in inception?

you know you didn't understand inception when...

5 understanding requirements

types of requirements

further readings

6 use-case model: writing requirements in context

goals and stories

background

use cases and adding value

use cases and functional requirements

use case types and formats

fully dressed example: process sale

explaining the sections

goals and scope of a use case

finding primary actors, goals, and use cases

congratulations: use cases have been written, and are

<<UML和模式应用>>

imperfect

- write use cases in an essential ui-free style
- actors
- use case diagrams
- requirements in context and low-level feature lists
- use cases are not object-oriented
- use cases within the up
- case study: use cases in the nextgen inception phase
- further readings
- up artifacts and process context
- 7 identifying other requirements
- nextgen pos examples
- nextgen example: (partial) supplementary specification
- commentary: supplementary specification
- nextgen example: (partial) vision
- commentary: vision
- nextgen example: a (partial) glossary
- commentary: glossary (data dictionary)
- reliable specifications: an oxymoron?
- online artifacts at the project website
- not much uml during inception?
- other requirement artifacts within the up
- further readings
- up artifacts and process context
- 8 from inception to elaboration
- checkpoint: what happened in inception?
- on to elaboration
- planning the next iteration
- iteration 1 requirements and emphasis: fundamental ooa/i)

skills

- what artifacts may start in elaboration?
- you know you didn't understand elaboration when...

Part iii elaboration iteration

- 9 use-case model: drawing system sequence diagrams
- system behavior
- system sequence diagrams
- example of an ssd
- inter-system ssds
- ssds and use cases
- system events and the system boundary
- naming system events and operations
- showing use case text
- ssds and the glossary
- ssds within the up
- further readings
- up artifacts
- 10 domain model: visualizing concepts

<<UML和模式应用>>

domain models
conceptual class identification
candidate conceptual classes for the sales domain
domain modeling guidelines
resolving similar conceptual classes--register vs. "post"
modeling the unreal world
specification or description conceptual classes
uml notation, models, and methods: multiple perspectives
lowering the representational gap
example: the nextgen pos domain model
domain models within the up
further readings
up artifacts
11 domain model: adding associations
associations
the uml association notation
finding associations-common associations list
association guidelines
roles
how detailed should associations be?
naming associations
multiple associations between two types
associations and implementation
nextgen pos domain model associations
nextgen pos domain model
12 domain model: adding attributes
attributes
uml attribute notation
valid attribute types
non-primitive data type classes
design creep: no attributes as foreign keys
modeling attribute quantities and unite
attributes in the nextgen domain model
multiplicity from saleslineitem to item
domain model conclusion
13 use-case model: adding detail with operation contracts
contracts
example contract: enteritem
contract sections
postconditions
discussion---enterItem postconditions
writing contracts leads to domain model updates
when are contracts useful? contracts vs. use cases?
guidelines: contracts
nextgen pos example: contracts
changes to the domain model
contracts, operations, and the uml

<<UML和模式应用>>

operation contracts within the up

further readings

14 from requirements to design in this iteration

iteratively do the right thing, do the thing right

didn't that take weeks to do? no, not exactly.

on to object design

15 interaction diagram notation

sequence and collaboration diagrams

example collaboration diagram: makepayment

example sequence diagram: makepayment

interaction diagrams are valuable

common interaction diagram notation

basic collaboration diagram notation

basic sequence diagram notation

16 grasp: designing objects with responsibilities

responsibilities and methods

responsibilities and interaction diagrams

patterns

grasp: patterns of general principles in assigning

responsibilities

the uml class diagram notation

information export (or expert)

creator

low coupling

high cohesion

controller

object design and crc cards

further readings

17 design model: use-case realizations with grasp patterns

use-case realizations

artifact comments

use-case realizations for the nextgen iteration

object design: makenewsale

object design: enteritem

object design: endsale

object design: makepayment

object design: startup

connecting the ui layer to the domain layer

use-case realizations within the up

summary

18 design model: determining visibility

visibility between objects

visibility

illustrating visibility in the uml

19 design model: creating design class diagrams

when to create dcdds

example dcd

<<UML和模式应用>>

- dcd and up terminology
- domain model vs. design model classes
- creating a nextgen pos dcd
- notation for member details
- dcds, drawing, and case tools
- dcds within the up
- up artifacts
- 20 implementation model: mapping designs to code
- programming and the development process
- mapping designs to code
- creating class definitions from dcds
- creating methods from interaction diagrams
- container/collection classes in code
- exceptions and error handling
- defining the sale--makelineitem method
- order of implementation
- test-first programming
- summary of mapping designs to code
- introduction to the program solution
- Part iv elaboration iteration
- 21 iteration 2 and its requirements
- iteration 2 emphasis: object design and patterns
- from iteration i to 2
- iteration 2 requirements
- refinement of analysis-oriented artifacts in this iteration
- 22 grasp: more patterns for assigning responsibilities
- polymorphism
- pure fabrication
- indirection
- protected variations
- 23 designing use-case realizations with gof design
- patterns
- adapter (gof)
- "analysis" discoveries during design: domain model
- factory (gof)
- singleton (gof)
- conclusion of the external services with varying interfaces
- problem
- strategy (gof)
- composite (gof) and other design principles
- facade (gof)
- observer/publish-subscribe/delegation event model (gof)
- conclusion
- further readings
- Part v elaboration iteration
- 24 iteration 3 and its requirements
- iteration 3 requirements

<<UML和模式应用>>

iteration 3 emphasis

25 relating use cases

the include relationship

terminology: concrete, abstract, base, and addition use

cases

the extend relationship

the generalize relationship

use case diagrams

26 modeling generalization

new concepts for the domain model

generalization

defining conceptual superclasses and subclasses

when to define a conceptual subclass

when to define a conceptual superclass

nextgen pos conceptual class hierarchies

abstract conceptual classes

modeling changing states

class hierarchies and inheritance in software

27 refining the domain model

association classes

aggregation and composition

time intervals and product prices--fixing an iteration 1

"error"

association role names

roles as concepts vs. roles in associations

derived elements

qualified associations

reflexive associations

ordered elements

using packages to organize the domain model

28 adding new ssds and contracts

new system sequence diagrams

new system operations

new system operation contracts

29 modeling behavior in statechart diagrams

events, states, and transitions

statechart diagrams

statechart diagrams in the up?.

use case statechart diagrams

use case statechart diagrams for the pos application

classes that benefit from statechart diagrams

illustrating external and interval events

additional statechart diagram notation

further readings

30 designing the logical architecture with patterns

software architecture

architectural pattern: layers

<<UML和模式应用>>

the model-view separation principle

further readings

31 organizing the design and implementation model packages

package organization guidelines

more uml package notation

further readings

32 introduction to architectural analysis and the sad

architectural analysis

types and views of architecture

the science: identification and analysis of architectural

factors

example: partial nextgen pos architectural factor table

the art: resolution of architectural factors

summary of themes in architectural analysis

architectural analysis within the up

further readings

33 designing more use-case realizations with objects and

patterns

failover to local services; performance with local caching

handling failure

failover to local services with a proxy (gof)

designing for non-functional or quality requirements

accessing external physical devices with adapters; buy vs.

build

abstract factory (gof) for families of related objects

handling payments with polymorphism and do it myself

conclusion

34 designing a persistence framework with patterns

the problem: persistent objects

the solution: a persistence service from a persistence

framework

frameworks

requirements for the persistence service and framework

key ideas

pattern: representing objects as tables

uml data modeling profile

pattern: object identifier

accessing a persistence service with a facade

mapping objects: database mapper or database broker pattern

framework design with the template method pattern

materialization with the template method pattern

configuring mappers with a mapperfactory

pattern: cache management

consolidating and hiding sql statements in one class

transactional states and the state pattern

designing a transaction with the command pattern

lazy materialization with a virtual proxy

<<UML和模式应用>>

- how to represent relationships in tables
- persistentobject superclass and separation of concerns
- unresolved issues
- Part vi special topics
 - 35 on drawing and tools
 - on speculative design and visual thinking
 - suggestions for uml drawing within the development process
 - tools and sample features
 - example two
 - 36 introduction to iterative planning and project issues
 - ranking requirements
 - ranking project risks
 - adaptive vs. predictive planning
 - phase and iteration plans
 - iteration plan: what to do in the next iteration?
 - requirements tracking across iterations
 - the (in)validity of early estimates
 - organizing project artifacts
 - some team iteration scheduling issues
 - you know you didn't understand planning in the up when...
 - further readings
 - 37 comments on iterative development and the up
 - additional up best practices and concepts
 - the construction and transition phases
 - other interesting practices
 - motivations for timeboxing an iteration
 - the sequential "waterfall" lifecycle
 - usability engineering and user interface design
 - the up analysis model
 - the rup product
 - the challenge and myths of reuse
 - 38 more uml notation
 - general notation
 - implementation diagrams
 - template (parameterized, generic) class
 - activity diagrams
 - bibliography
 - glossary
- index

<<UML和模式应用>>

编辑推荐

《UML和模式应用:面向对象分析和设计及统一过程导论(第2版)(影印版)》：在这一版中，Larman保持了其一贯精准和细心的写作风格，这是一本比第一版更好的书。很少有人能掌握解释事情的诀窍，而能够很好地对软件分析和设计进行解释的人则更少，但Craig Larman做到了这两点。

<<UML和模式应用>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>