



UML Distilled Third Edition

UMLモデリングのエッセンス 第3版

標準オブジェクトモデリング言語入門

マーチン・ファウラー 著

羽生田栄一 監訳



UML Distilled Third Edition

UMLモデリングのエッセンス 第3版

標準オブジェクトモデリング言語入門

マーチン・ファウラー 著

羽生田栄一 監訳

江苏工业学院图书馆
藏书章

■著者紹介

Martin Fowler (マーチン・ファウラー)

ThoughtWorks社のチーフサイエンティスト。10年以上にわたりオブジェクト指向テクノロジーをエンタープライズ系のソフトウェア開発に適用してきた。パターン、UML、リファクタリング、およびアジャイル手法の第一人者。マサチューセッツ州メルローズ在住。

【著書】

- ・ *Analysis Patterns: Reusable Object Modeling*, 1997
『アナリシスパターン』、発行：アジソン・ウェスレイ・パブリッシャーズ・ジャパン、発売：星雲社、1998
- ・ *Refactoring: Improving the Design of Existing Code*, 1999
『リファクタリング』、ピアソン・エデュケーション、2000
- ・ *Patterns of Enterprise Application Architecture*, 2002
『エンタープライズアプリケーションアーキテクチャパターン』、翔泳社、2005

【ホームページ】

<http://www.martinfowler.com/>

装丁	河原田 智 (ポルターハウス)
編集・組版	有限会社 風工舎
翻訳	多摩ソフトウェア 有限会社

ユー・エム・エル

UML モデリングのエッセンス 第3版

2005年 6月13日 初版第1刷発行

2006年 6月15日 初版第3刷発行

著 者	マーチン・ファウラー
監 訳 者	羽生田 栄一
発 行 人	速水 浩二
発 行 所	株式会社 翔泳社 (http://www.seshop.com/)
印刷・製本	株式会社 廣済堂

本書は著作権法上の保護を受けています。本書の一部または全部について（ソフトウェアおよびプログラムを含む）、株式会社 翔泳社から文書による許諾を得ずに、いかなる方法においても無断で複写、複製することは禁じられています。

本書へのお問い合わせについては、iiページに記載の内容をお読みください。

落丁・乱丁はお取り替えいたします。03-5362-3705までご連絡ください。

ISBN4-7981-0795-6

Printed in Japan

本書内容に関するお問い合わせについて

このたびは翔泳社の書籍をお買い上げいただき、誠にありがとうございます。弊社では、読者の皆様からのお問い合わせに適切に対応させていただくため、以下のガイドラインへのご協力をお願いしております。下記項目をお読みいただき、手順に従ってお問い合わせください。

●お問い合わせの前に

弊社Webサイトの「正誤表」や「出版物Q&A」をご確認ください。これまでに判明した正誤や追加情報、過去のお問い合わせへの回答(FAQ)、的確なお問い合わせ方法などが掲載されています。

正誤表 <http://www.seshop.com/book/errata/>

出版物Q&A <http://www.seshop.com/book/qa/>

●ご質問方法

弊社Webサイトの書籍専用質問フォーム(<http://www.seshop.com/book/qa/>)をご利用ください(お電話や電子メールによるお問い合わせについては、原則としてお受けしておりません)。

※質問専用シートのお取り寄せについて

Webサイトにアクセスする手段をお持ちでない方は、ご氏名、ご送付先(ご住所/郵便番号/電話番号またはFAX番号/電子メールアドレス)および「質問専用シート送付希望」と明記のうえ、電子メール(qaform@shoeisha.com)、FAX、郵便(80円切手同封)のいずれかにて「編集部読者サポート係」までお申し込みください。お申し込みの手段によって、折り返し質問シートをお送りいたします。

シートに必要な事項を漏れなく記入し、「編集部読者サポート係」までFAXまたは郵便にてご返送ください。

●回答について

回答は、ご質問いただいた手段によってご返事申し上げます。ご質問の内容によっては、回答に数日ないしはそれ以上の期間を要する場合があります。

●ご質問に際してのご注意

本書の対象を越えるもの、記述箇所を特定されていないもの、また読者固有の環境に起因するご質問等にはお答えできませんので、予めご了承ください。

●郵便物送付先およびFAX番号

送付先住所 〒160-0006 東京都新宿区舟町5
FAX番号 03-5362-3818
宛先 (株)翔泳社 出版局 編集部読者サポート係

.....
※本書に記載されたURL等は予告なく変更される場合があります。

※本書の出版にあたっては正確な記述につとめました。著者や出版社などのいずれも、本書の内容に対してなんらかの保証をするものではなく、内容やサンプルに基づくいかなる運用結果に関してもいっさいの責任を負いません。

※本書に掲載されているサンプルプログラムやスクリプト、および実行結果を記した画面イメージなどは、特定の設定に基づいた環境にて再現される一例です。

.....
※本書に記載されている会社名、製品名などは、一般に各社の商標または登録商標です。

※本書では®および™は割愛させていただいております。

UML Distilled Third Edition: A Brief Guide to the Standard Object Modeling Language by Martin Fowler

Copyright © 2004 by Pearson Education, Inc.

Japanese translation rights arranged with Pearson Education, Inc., through Japan UNI Agency, Inc., Tokyo.

古代から、最も優れた建築家や設計者は節約の重要性を心得ていました。「過ぎたるは及ばざるがごとし」という逆説や「禅の精神は初心者精神である」という公案に表されたこの知恵は永遠のものです。つまり、無駄を最大限まで省くと、その形状は機能と調和します。ピラミッドからシドニーオペラハウスに至るまで、あるいはノイマン型アーキテクチャからUNIXやSmalltalkに至るまで、最高の建築家や設計者はこの普遍的で永遠の原則に従おうと努力してきました。

私はオッカムの剃刀の価値を認識している者として、設計や読書の際に節約の法則に従ったプロジェクトや本を求めてきました。その私にとって、今皆さんが手にされているこの本を心からお勧めするのは当然のことです。

この言葉に最初は驚かれるかもしれません。私は、UML (Unified Modeling Language) を定義する膨大で詳細な仕様にたびたび関わっています。ツールベンダーはこれらの仕様を使ってUMLを実装し、方法論者たちはそれを適用します。7年の間、私はUML 1.1とUML 2.0を定義および改訂する大規模な国際標準化チームの議長を務めてきました。この間に、UMLは表現や精度という面では成熟しましたが、標準化の結果として複雑なものになってしまいました。残念なことに、標準化はUMLの無駄を省いて簡潔の実現することではなく、委員会による設計上の妥協であるとみなされています。

詳細で難解な仕様に慣れているUMLの専門家が、著者が抽出したUML 2.0の「エッセンス」から何を学べるのでしょうか？ 極めて多くのことを学ぶことができるでしょう。それは読者の皆さんも同じです。最初に、著者は複雑なUMLから多くの部分を省き、彼の経験から効率的であると証明された実用的な部分だけを抽出しています。また、今回の最新版を出版するに当たって、著者は単純にページを追加するのではなく、全面的な書き直しを行っています。UMLが発展しても、著者は「最も実用的なUML」を探し、それについてのみ説明するという目的に誠実であろうとしています。著者が説明するUMLは全体の20パーセントにすぎませんが、それで実務の80パーセントはカバーできます。この困難な作業をやり遂げたことは、素晴らしい功績です。

さらに印象的なのは、著者が魅力的な文章で説明していることです。著者の意見とエピソードが公開されており、本書は読み物としても楽しいものになっています。また、シ

システムの設計はクリエイティブかつ生産的であるべきだということを私たちに思い出させてくれます。節約の公案を常に心掛けていれば、小学校での絵の具遊びや絵画のクラスが楽しかったように、UMLによるモデリング作業が楽しいものであることがわかります。UMLは、私たちのクリエイティビティを誘導するためのツールであると同時に、システムの仕様を正確に表す設計図としてサードパーティーがシステムの入札や設計に使用できる必要があります。これは、どんな設計用の言語にも適用すべき普遍的なテストと言えます。

本書は薄い本ですが、同時に重みのある本です。UML 2.0の説明から多くを学べるだけでなく、著者のモデリングアプローチからも多くを学ぶことができます。

この版で説明されているUML 2.0の言語機能を選び出し、その正確性を高める作業を著者とともに行ったのは、有意義な体験でした。自然言語であれ人工言語であれ、使用されているすべての言語は進化するか消滅するものであるということを常に心に留めておく必要があります。UMLの改訂作業では、著者が選択した新しい機能や読者のような現場のユーザーがよく使用する機能が考慮されています。実際に使用されることによって言語は生き残り、市場で自然淘汰されることによって言語は進化します。

モデル駆動型開発が主流になるまでには困難な作業が必要ですが、UMLモデリングの基本を明確に説明し、プログラムに反映させている本書には勇気を与えられます。私と同様に、読者の皆様が本書から多くを学び、新しい見識をもってソフトウェアモデリング作業に臨まれることをお祈りします。

Cris Kobryn

U2 Partners' UML 2.0 Submission Team議長

Telelogic、チーフ・テクノロジスト

推薦のことば (初版)

私たちがUML (Unified Modeling Language) の開発を始めた時点で考えていた目標は、ソフトウェアシステムのモデリングプロセスの霧を払って明解なものにし、産業界の最高の技術を反映した設計を表現するための標準的な手段を示すことでした。標準的なモデリング言語があれば、ソフトウェアを構築する前にモデルを作成する開発者が増えるはずで、開発者の間では、標準的なモデリング言語の有効性は周知の事実であり、UMLが急速に広まりつつあります。

UMLを作り上げることは、それ自体が繰り返しと積み重ねで、大規模なソフトウェアシステムのモデルの作成と非常に似ていました。最終的に目指しているものは、オブジェクト指向に携わる多くの個人や企業によって生み出されたさまざまな発想や提案を反映した標準です。UMLの作成に着手したのは私たちですが、これを優れた技法として完成できたのは他の多くの人々のおかげであり、その協力に感謝しています。

意見をまとめて標準モデリング言語を作成することは、それ自体、意義があります。また、開発コミュニティを教育し、ソフトウェア開発プロセスの中で利用しやすいUMLを提供することも重要です。最新のUMLについて簡潔にまとめられた本書の中で、Martin Fowler氏は、こうした取り組みに対して十分以上の成果を上げています。

本書では、UMLの主要な部分が明確にわかりやすく説明されます。また、開発プロセスでUMLが果たす役割が明瞭に示されます。UMLを完成させる過程で大きな助けとなったのは、Martin氏の12年以上にも及ぶ設計とモデリングの経験から引き出される洞察力と知識でした。

このようなMartin氏によって著された本書は、開発者にUMLの利点を示し、モデリングを追究する意欲をかき立てます。

初めてUMLに取り組む方や、開発プロセスでUMLが果たす役割を全体的に把握したいと考えるモデリング担当者と開発者に、本書をお勧めします。

Grady Booch
Ivar Jacobson
James Rumbaugh

人生において、私はいろいろな意味でラッキーでした。最初の幸運は、1997年に本書の初版を適切なタイミングで正しい知識に基づいて書くことができたことです。当時は、混沌としていたオブジェクト指向（Object Oriented：OO）モデリングの世界がUML（Unified Modeling Language）に統合されようとしていたときでした。その後、UMLはオブジェクトだけでなく、ソフトウェア一般をグラフィカルにモデリングするための標準になりました。私の幸運は、本書が25万部以上も売れて、UML業界で最も一般的な本になったことです。

これは私にとっては良いことですが、本書を買う必要があるかどうかはわかりません。

強調したいのは、本書がUMLの概要について述べた本であるという点です。つまり、本書は長年にわたって発達してきたUMLのすべての機能について詳細に説明することを目的としていません。本書は、最も実用的なUMLを抽出し、それらに絞って解説することを目的としています。UMLについて詳細に説明した厚い本もありますが、読むには長い時間がかかります。読書に費やす時間というのは大きな投資です。この薄い本では、最も実用的なUMLだけが選択されています。そのため、読者の皆さんは自分でUMLを選択する手間を省くことができます。ただ、残念ながら本が薄くなっても値段が比例して安くなるわけではありません。質の高い技術書を出版するには、一定の費用がかかります。

本書の目的のひとつは、UMLについて学習することです。この本は薄いので、UMLの要点を短時間で把握できます。UMLを熟知している設計者の方は、『UMLユーザーガイド』（Booch 他 [6]）や『UMLリファレンスマニュアル』（Rumbargh [40]）などの分厚い本で詳細をご確認ください。

また、本書はUMLの最も一般的な部分に関するリファレンスとして手軽に活用できます。すべての項目については説明していませんが、他のUMLの本よりも軽いため、持ち運びも楽です。

また、これは特殊な本でもあります。私は長い間オブジェクト指向を扱ってきました。そのため、どの機能が実用的であるかという点について、具体的な考えを持っています。どの本にも著者の見解が反映されていますが、私も自分の見解を隠さないようにしました。そのため、客観的な意見が必要な場合は他の本を読まれることをお勧めします。

本書はオブジェクト指向の良い入門ガイドであると言われることがありますが、本来の目的は異なります。オブジェクト指向設計の基礎について知りたい場合は、Craig Larman氏の『実践UML』[29]をお勧めします。

UMLに興味がある設計者の多くはツールを使用します。本書はUMLの標準的用法と慣例的用法を対象にしており、さまざまなツールが何をサポートしているかという詳細については説明していません。UMLの登場によって、表記に関するそれまでの膨大な問題が解決されました。ただし、ツールによってUMLのダイアグラムの描画方法がさまざまに異なるという点が面倒です。

本書では、MDA (Model Driven Architecture : モデル駆動型アーキテクチャ) の説明にはページを割いていません。多くの人がMDAとUMLを同じものだと考えていますが、開発者の多くはUMLを使用するだけでMDAには興味を持っていません。MDAを習得したい場合は、本書でUMLの概要を理解した上で、MDAについて詳しく説明している本をお読みになることをお勧めします。

本書ではUMLに焦点を当てていますが、他の関連する技法についても紹介しています。たとえば、オブジェクト指向設計に有益なCRCカード法などです。UMLはオブジェクトを扱う際の手段のひとつにすぎません。そのため、他の技法を導入することも重要です。

この薄い本では、UMLとソースコードの関連性について詳細に説明することはできません。これは、UMLとソースコードを対応させる標準的な方法がないことも大きな理由です。ただし、UMLを実装するための一般的なコード記述技法については説明しています。本書では、JavaとC#のコード例を紹介しています。これは、一般的にこの2つの言語が最も広く理解されているためです。これらが私のお気に入りの言語であるというわけではありません。実際には、私は長い間Smalltalkを使用してきました。

UMLを使用する理由

私たちが設計をグラフィカルに表現するようになって、ある程度の時間が経っています。私がこの記法を使用するのは、これがコミュニケーションと理解に役立つからです。良いダイアグラムは、設計に関するアイデアを出し合う際に役立ちます。特に、大まかな設計について話し合う際に有効です。また、ダイアグラムを使うと、ソフトウェアシステムやビジネスプロセスを理解しやすくなります。設計チームの一員として作業している場合、ダイアグラムは設計の理解に役立つだけでなく、設計についてチームで話し合うときにも有効です。ダイアグラムは、文字で表されたプログラミング言語の代わりに使用することはできませんが、その補助としては役立ちます。

多くの人は、将来、グラフィカルな技法がソフトウェア開発において主要な役割を担うようになると信じています。私はその考えに懐疑的なほうですが、グラフィカルな表現を

使用してできることとできないことを正しく把握することは必要です。

いくつかあるグラフィカルな記法の中でもUMLが重要であるのは、それがオブジェクト指向開発のコミュニティで広く使用されており標準化されているからです。UMLは、オブジェクト指向における主要なグラフィカル表記となっただけでなく、オブジェクト指向以外でも一般的な技法となっています。

本書の構成

第1章では、UMLの概要を説明します。UMLとは何か、UMLが各種のユーザーにとって持つさまざまな意味、UMLの由来を説明します。

第2章では、ソフトウェアプロセスを取り上げます。プロセスとUMLはまったく別の存在ですが、UMLのようなものの場合は、そのコンテキストを理解するためにプロセスを理解することが必要です。特に、反復型開発の役割を理解することは重要です。反復型開発は、ほとんどのオブジェクト指向コミュニティにおいてプロセスの基本的なアプローチでした。

残りの章では、UML内で使用するダイアグラムの種類について説明します。第3章と第4章では、UMLの最も実用的なダイアグラムであるクラス図（コア）とシーケンス図について説明します。ページ数はそれほどでもありませんが、これらの章で説明されている技法を使用することによって、UMLの機能を最大限に活用できると信じています。UMLは巨大で進化し続ける技法ですが、そのすべてを利用する必要はありません。

第5章では、クラス図の機能のうち、必要性は低いものの便利なものについて説明します。第6章から第8章では、システムの構造を明らかにする3つの便利なダイアグラムであるオブジェクト図、パッケージ図、および配置図について説明します。

第9章から第11章では、3つの便利な振る舞いの技法であるユースケース、状態図（正式には「状態マシン図」と言いますが、しばしば「状態図」と呼ばれます）、およびアクティビティ図について説明します。第12章から第17章では、一般的にあまり重要でないダイアグラムについて、簡単な説明と短い例を使ってごく簡単にまとめています。

本書の見返しには、最も重要な記法がまとめられています。これまで、この見返しが本書の最も重要な部分であるという話をたびたび耳にしてきました。本書の他の部分をお読みになるときに、この部分を参照すると便利です。

第3版の変更点

本書の旧版をお持ちの方は、前の版からの変更点と、新しい版を購入する必要があるかどうかをお知りになりたいだろうと思います。

第3版を出版するきっかけとなったのは、UML 2が登場したことでした。UML 2には、新しい技法が多数追加されています。たとえば、新しい種類のダイアグラムなどです。シーケンス図の相互作用フレームなど、これまでのダイアグラムにも新しい表記が多数追加されました。UML 2における変更点を理解したいが（UMLの膨大な）仕様を確認するのは避けたい（私もお勧めしません）という場合に本書は最適です。

また、第3版ではほとんどの部分を書き直し、本文と例を最新のものに変更しました。過去5年間にわたり、UMLを教えたり使用することで学んだ知識も盛り込みました。したがって、この極めて薄い本は今までの精神を保ちつつ、ほとんどの内容は新しくなっています。

過去数年間、私は本書に最新の情報を収めようと努力してきました。UMLが変更されるたびに、それに対応するように最善を尽くしてきました。本書は、2003年6月に担当委員会によって承認されたUML 2の草案に基づいて書かれています。今後、正式版が発表されるまでにUML 2が変更される可能性は低いようです。そこで、本書の改訂版を出版する機は熟したと判断しました。更新情報については、私のWebサイト (<http://martinfowler.com>) をご覧ください。

謝辞

本書を執筆する数年の間に、たくさんの方からの支援を受けました。最初に、Carter Shanklin氏とKendall Scott氏に感謝します。Carter氏はアジソン・ウェスレイ社の編集者で、本書の執筆を勧めてくれました。Kendall Scott氏は、最初の2つの版の本文と図をまとめるのを手伝ってくれました。Carter氏とScott氏は、アジソン・ウェスレイ社の名に恥じない品質を保ちながら、初版を驚くほど短時間で出版しました。また、UML初期のまだ何も安定していないころ、変更がでるたびに原稿を修正してくれました。

Jim Odell氏は、私のキャリアが始まったところからのよき相談相手です。彼は、頑固な方法論者を納得させ、共通の標準に同意させる際の技術的および人間面的問題に深く関わっていました。Odell氏の本書に対する貢献は計り知れず、深くお礼を申し上げます。また、それは本書だけでなく、UMLに対する貢献についても同じです。

UMLは標準から生まれたものですが、私は標準化団体が苦手です。そのため、UMLの最新情報を得るためには、委員会の最新の動きを私に知らせてくれる「スパイ」のネットワークが必要です。Conrad Bock氏、Steve Cook氏、Cris Kobryn氏、Jim Odell氏、

Guus Ramackers氏、およびJim Rumbaugh氏の「スパイ活動」なしには何もできませんでした。彼らは私に助言を与え、愚かな質問にも答えてくれました。

Grady Booch氏、Ivar Jacobson氏、およびJim Rumbaugh氏は「three amigos」として知られています。数年にわたって私がこのようなふざけた名前で呼んでいるにもかかわらず、彼らは本書の執筆をサポートし、私を勇気づけてくれました。悪ふざけは私の愛情表現なのです。

良書を出版するには、優れた校閲者が不可欠です。私はCarter氏から、校閲者は多すぎないほうがよいということを学びました。本書の旧版の校閲者は、Simmi Kochhar Bhargava氏、Grady Booch氏、Eric Evans氏、Tom Hadfield氏、Ivar Jacobson氏、Ronald E. Jeffries氏、Joshua Kerievsky氏、Helen Klein氏、Jim Odell氏、Jim Rumbaugh氏、およびVivek Salgar氏でした。

次に、第3版を担当した優秀な校閲者を紹介します。

Conrad Bock氏	Craig Larman氏
Andy Carmichael氏	Steve Mellor氏
Alistair Cockburn氏	Jim Odell氏
Steve Cook氏	Alan O'Callaghan氏
Luke Hohmann氏	Guus Ramackers氏
Pavel Hruby氏	Jim Rumbaugh氏
Jon Kern氏	Tim Seltzer氏
Cris Kobryn氏	

上記の校閲者は時間を割いて原稿を読み、それぞれが少なくとも1つの恥ずかしい間違いを見つけてくれました。心から感謝いたします。本書に残っている間違いは、すべて私の責任です。私が間違いを見つけたときは、martinfowler.comの「Books」セクションにある正誤表に記入します。

UML仕様を設計および記述したチームのメンバーは、Don Baisley氏、Morgan Björkander氏、Conrad Bock氏、Steve Cook氏、Philippe Desfray氏、Nathan Dykman氏、Anders Ek氏、David Frankel氏、Eran Gery氏、Øystein Haugen氏、Sridhar Iyengar氏、Cris Kobryn氏、Birger Møller-Pedersen氏、James Odell氏、Gunnar Övergaard氏、Karin Palmkvist氏、Guus Ramackers氏、Jim Rumbaugh氏、Bran Selic氏、Thomas Weigert氏、およびLarry Williams氏です。彼らの協力がなければ、何も書くことはできなかったはずです。

UMLダイアグラムの作成には、Pavel Hruby氏が開発したいくつかの優れたVisioテンプレートを使用しました。これらのテンプレートは、<http://phruby.com>で手に入れることができます。

多くの人がネット上でまたは実際に私と話をし、助言や質問をしてくださったり間違いを指摘してくださいました。すべての人を挙げることはできませんが、心から感謝しています。

マサチューセッツ州、バーリントンにあるSoftproという技術書専門店のスタッフの皆さんに感謝いたします。彼らが何時間も在庫を読ませてくれたおかげで、ユーザーが日ごろUMLをどのように使用しているかがわかりました。また、本を読んでいる間に、おいしいコーヒーを出してくれました。

第3版の編集者はMike Hendrickson氏です。Kim Arney Mulcahy氏はプロジェクトを管理するだけでなく、レイアウトやダイアグラムを整えてくれました。John Fuller氏はアジソン・ウェスレイ社制作部門の編集者です。また、Evelyn Pyle氏とRebecca Rider氏は原稿を整理し校正してくれました。皆様に心から感謝いたします。

本書の執筆中、Cindyはいつも側にいてくれました。今、彼女は庭仕事をしています。両親は私に良い教育を受けさせてくれました。それがすべての原点となっています。

Martin Fowler
マサチューセッツ州、メルローズ
<http://martinfowler.com>

推薦のことば（第3版）	iii
推薦のことば（初版）	v
著者まえがき	vii

第1章 UMLの概要	1
1.1 UMLとは	1
1.2 UMLの使い方	2
1.3 UMLを理解するには	7
1.4 記法とメタモデル	9
1.5 UMLダイアグラム	11
1.6 正しいUMLとは	13
1.7 UMLの意味	14
1.8 UMLだけでは不十分	15
1.9 UMLのどこから手を付けるか	16
1.10 さらに詳しい情報を得るには	16

第2章 開発プロセス	19
2.1 反復型プロセスとウォーターフォール型プロセス	19
2.2 予測的計画と適応的計画	23
2.3 アジャイルプロセス	25
2.4 Rational Unified Process	25
2.5 プロセスをプロジェクトに適合させる	26
2.6 UMLをプロセスに適合させる	29
2.6.1 要求分析	29
2.6.2 設計	30
2.6.3 文書化	31
2.6.4 レガシーコードを理解する	33

2.7	開発プロセスの選択	33
2.8	さらに詳しい情報を得るには	33

第3章 クラス図：基本的要素 35

3.1	プロパティ	35
3.1.1	属性	36
3.1.2	関連	37
3.2	多重度	38
3.3	プロパティのプログラミング的解釈	39
3.4	双方向関連	41
3.5	操作	44
3.6	汎化	45
3.7	ノートとコメント	46
3.8	依存関係	47
3.9	制約規則	49
3.10	いつクラス図を使用すべきか	52
3.11	さらに詳しい情報を得るには	52

第4章 シーケンス図 53

4.1	参加者の生成と削除	56
4.2	ループ、条件など	57
4.3	同期的な呼び出しと非同期の呼び出し	61
4.4	いつシーケンス図を使用すべきか	61

第5章 クラス図：上級概念 65

5.1	キーワード	65
5.2	レスポンスビリティ (責務)	66
5.3	静的 (static) な操作および属性	66
5.4	集約とコンポジション	67
5.5	派生プロパティ	68
5.6	インターフェイスと抽象クラス	69
5.7	読み取り専用 (readOnly) と凍結 (frozen)	72
5.8	参照オブジェクトと値オブジェクト	73
5.9	限定子付き関連	75
5.10	分類と汎化	76
5.11	多重分類と動的分類	77
5.12	関連クラス	78
5.13	テンプレート (パラメータ化された) クラス	81
5.14	列挙	83
5.15	アクティブクラス	83

5.16	可視性	84
5.17	メッセージ	84
第6章	オブジェクト図	87
6.1	いつオブジェクト図を使用すべきか	87
第7章	パッケージ図	89
7.1	パッケージと依存関係	91
7.2	パッケージのアスペクト	93
7.3	パッケージの実装	94
7.4	いつパッケージ図を使用すべきか	95
7.5	さらに詳しい情報を得るには	95
第8章	配置図	97
8.1	いつ配置図を使用すべきか	98
第9章	ユースケース	99
9.1	ユースケースの内容	100
9.2	ユースケース図	102
9.3	ユースケースのレベル	103
9.4	ユースケースとフィーチャー（またはストーリー）	104
9.5	いつユースケースを使用すべきか	104
9.6	さらに詳しい情報を得るには	105
第10章	状態マシン図	107
10.1	内部アクティビティ	109
10.2	アクティビティ状態	109
10.3	スーパー状態	111
10.4	並行状態	111
10.5	状態図（状態マシン図）の実装	112
10.6	いつ状態図を使用すべきか	114
10.7	さらに詳しい情報を得るには	115
第11章	アクティビティ図	117
11.1	アクションの分解	119
11.2	区画（partition）	120

11.3	シグナル	121
11.4	トークン	123
11.5	フローとエッジ	124
11.6	ピンと変換	125
11.7	拡張領域	126
11.8	フロー終了	127
11.9	ジョイン仕様	128
11.10	さらに	128
11.11	いつアクティビティ図を使用すべきか	129
11.12	さらに詳しい情報を得るには	129
第12章	コミュニケーション図	131
12.1	いつコミュニケーション図を使用すべきか	133
第13章	コンポジット構造	135
13.1	いつコンポジット構造を使用すべきか	136
第14章	コンポーネント図	139
14.1	いつコンポーネント図を使用すべきか	141
第15章	コラボレーション	143
15.1	いつコラボレーションを使用すべきか	145
第16章	相互作用概要図	147
16.1	いつ相互作用概要図を使用すべきか	147
第17章	タイミング図	149
17.1	いつタイミング図を使用すべきか	149
付録A	UMLのバージョン間の相違点	151
A.1	UMLの改訂	151
A.2	本書の変更点	152
A.3	UML 1.0から1.1への変更点	153
A.3.1	タイプと実装クラス	153
A.3.2	汎化の弁別子に対するcomplete制約の解釈	153