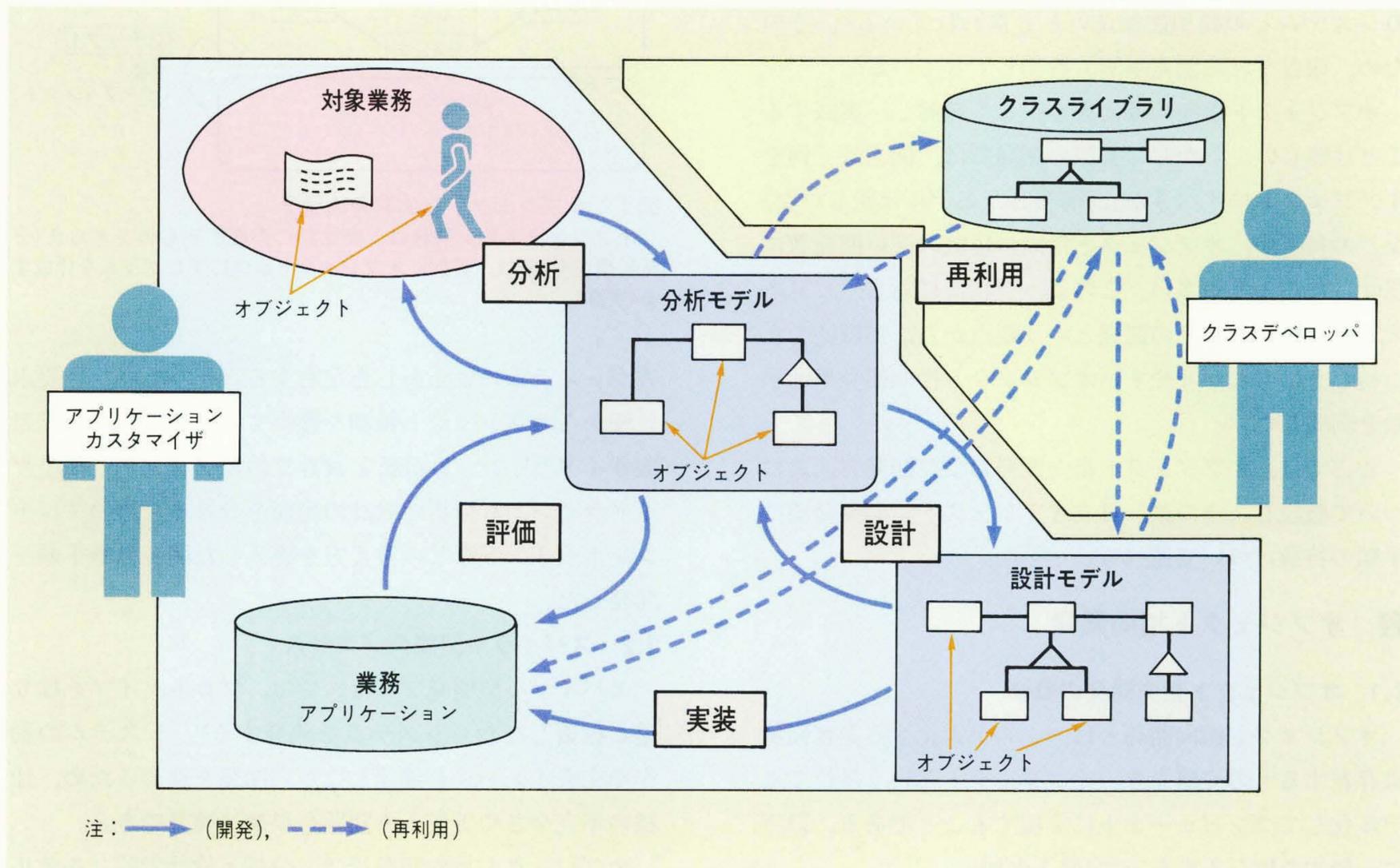


# オブジェクト指向技術を用いたソフトウェアの開発技法

New Methodology for System Development Using Object-Oriented Technology

千葉寛之\* Hiroyuki Chiba 高舘公人\*\*\* Masato Takadachi  
初田賢司\*\* Kenji Hatsuda 葉木洋一\* Yōichi Hagi  
竹田 滋\*\* Shigeru Takeda



## オブジェクト指向開発

オブジェクト指向開発は、アプリケーションの骨格や汎(はん)用部品を高度にモデル化、精練して提供するクラスデベロッパと、対象業務を分析し、それに合わせてシステムを作り上げるアプリケーションカスタマイザの共同作業により、質の高いシステムを効率よく構築することが可能となる。

オブジェクト指向技術は、人や書類など目にみえる「もの(オブジェクト)」に対応したソフトウェア部品を作り、それを組み合わせることで、保守性の良い情報システムを構築する技術である。しかし、その効果を十分に引き出すためには、システム化の対象業務をよく理解し、上手に「オブジェクト」をとらえる必要がある。そこで、過去の分析・設計の経験から得た「オブジェクト」のとらえ方のノウハウをわかりやすい手順として制定した「日立オブジ

ェクト指向開発標準手順」を開発した。

この標準手順の特徴は、ユーザーの要求を段階的に分析・設計に反映する「スパイラルアプローチ」と呼ぶ開発法を取り入れた点にある。これにより、ユーザー要求を的確に把握し、かつ保守性の良いソフトウェア部品構成が作成でき、高品質なシステム構築が容易になった。

また、この標準手順の実施を支援するツール群もあわせて開発し、製品化した。

\* 日立製作所 ビジネスシステム開発センター \*\* 日立製作所 情報システム事業部 \*\*\* 日立製作所 システム開発研究所

## 1 はじめに

オブジェクト指向の考え方で構築されたシステムは理解しやすく、かつ拡張性が高いという特徴があり、その考え方はCSS(Client Server System)すなわちクライアントとサーバが通信しながら所望の処理を実現する形態のシステムとの親和性が良いとも言われている<sup>1)</sup>。そのため、報告される開発事例も増加してきている<sup>2)</sup>。

オブジェクト指向の考え方を正しく理解し、実践することは難しい。しかし、実際の開発では、例えば「何をオブジェクトにすべきか」、「構築したモデルは正しいか」などの判断が、オブジェクト指向の経験の浅い開発者に任せられることが多い。このような状況に対し、ビジネスアプリケーションの開発という観点から、実践的かつ初心者にもわかりやすいオブジェクト指向開発標準手順を開発した。

ここでは、オブジェクト指向開発および開発方法論について概説し、その後、日立オブジェクト指向開発標準手順の特徴について述べる。

## 2 オブジェクト指向開発

### 2.1 オブジェクト指向開発の概要

オブジェクト指向開発とは、システム化する業務領域に存在するものや概念を、オブジェクトという単位でモデル化してコンピュータ上に実現することである。以下に一般的な開発手順を示す(図1参照)。

#### (1) ステップ1：要求定義

開発するシステムに要求される業務上の仕様を決定する。

#### (2) ステップ2：分析

対象業務領域のオブジェクトを抽出し、それらの間の関係を明らかにするオブジェクトモデル、オブジェクトの動作や反応を規定する動的モデル、およびオブジェクトへの入力から出力への変換を表現する機能モデルを構築する。

#### (3) ステップ3：設計

分析で得られたオブジェクトをコンピュータ上に実現するために、オブジェクトモデル、動的モデル、および機能モデルを詳細化する。さらに、ユーザーインターフェースやデータベースなどを接続するための構造を決定する。

#### (4) ステップ4：実装

設計の結果をプログラムに変換する。

オブジェクト指向開発では、「もの」という単位のと

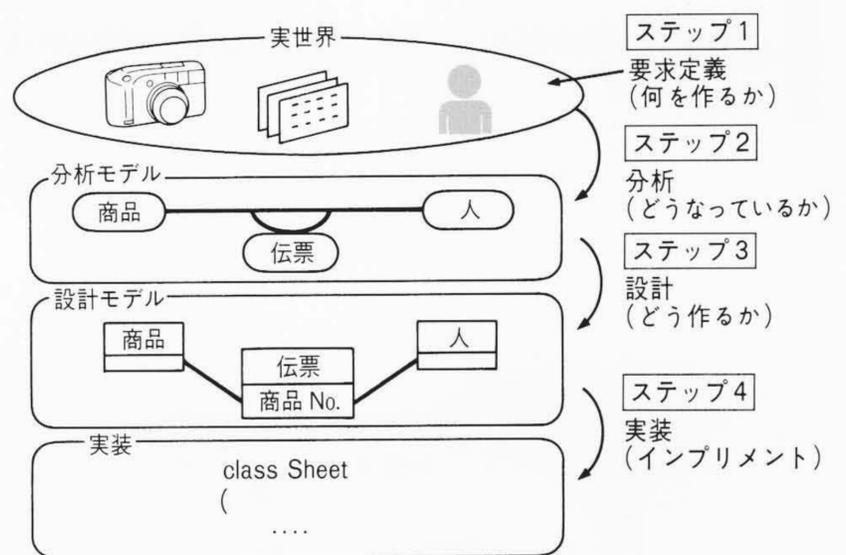


図1 オブジェクト指向開発の概要

オブジェクト指向開発は、実世界に存在するものにとらえ(分析)、構造を整理し(設計)、オブジェクト単位にプログラムを作成する(実装)。

方がシステムのよしあしを左右する。そのため、(1) 要求仕様からオブジェクト候補を獲得するための要求仕様定義ガイドライン、(2) 分析と設計でのオブジェクト抽出ガイドライン、(3) 分析と設計の範囲を分けるためのマルチコンポーネントなどの考え方を導入した開発標準手順を開発した。

### 2.2 スパイラル型開発プロセス

スパイラル型開発プロセスでは、プロトタイプを繰り返し拡張しながらシステムを開発する<sup>3)</sup>。システムの動作や完成イメージを確認しながら開発を進めるため、仕様の不完全さや実装上の問題を早期に発見できる。

オブジェクト指向開発では、分析と設計で同じモデルを使用する。したがって、プロトタイピングを繰り返しながら開発するスパイラル型開発プロセスに適している。

オブジェクト指向開発を、スパイラル型開発プロセスで実施すると、「ステップ1」から「ステップ4」を何回か繰り返す。このとき、何回繰り返せばよいのか、どの時点で終了すればよいのか、その判断基準の明確化が必要である。そこで、ステージという考え方を導入し、進捗(ちよく)管理を考慮した開発プロセスを開発した。

### 2.3 再利用

オブジェクト指向開発では、すでに開発されたオブジェクトを再利用することにより、新規開発部分が少なくて済み、信頼性の高いシステムを短期間で開発することが可能となる。再利用可能なオブジェクト部品を集めたものをクラスライブラリと呼ぶ。

再利用による効果をあげるためには、クラスライブラリの整備が不可欠であり、クラスライブラリを精練、利用するための手順も必要である。

### 3 日立オブジェクト指向開発標準手順

#### 3.1 日立オブジェクト指向開発標準手順の構成

クラスライブラリの開発では、高度なモデリングの技術が要求される。一方、アプリケーションの開発では、クラスライブラリの利用技術が要求される。これらの技術は性質を異にするものである。日立オブジェクト指向開発標準手順では、開発者の役割を三種類に分類し、おのおのの役割に応じた標準手順を提供する(図2参照)。

##### (1) クラスデベロッパ

クラスライブラリの開発・精練を担当するモデリングのスペシャリストである。すでに開発されたアプリケーションからクラスを抽出し、精練する。

##### (2) アプリケーションカスタマイザ

クラスライブラリを対象業務に適するようにカスタマイズして、業務アプリケーションを開発する。

##### (3) プロジェクト管理者

開発プロジェクトの計画や見積もりをたて、プロジェクトの進捗管理を行う。

クラスデベロッパとアプリケーションカスタマイザが協力することで、クラスライブラリの整備と再利用が促進される(図3参照)。

#### 3.2 要求仕様定義ガイドライン

多くのオブジェクト指向方法論が対象としている開発プロセスは、システムに関する記述がすでに存在しているところから始まっている。しかし、システムに関する記述を獲得すること、つまり要求仕様を定義することが不十分であれば、分析が不十分になることは明らかである。

クラスデベロッパとアプリケーションカスタマイザ向

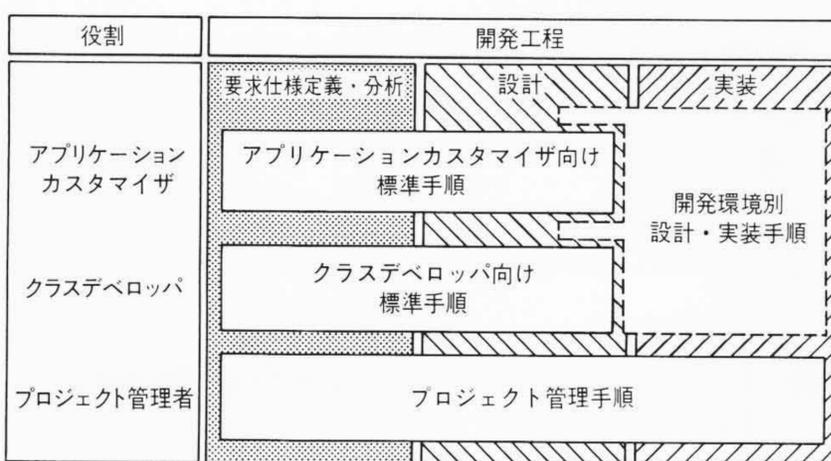


図2 オブジェクト指向開発標準手順の構成

実際にクラスライブラリを再利用して開発を行うアプリケーションカスタマイザと、既存のオブジェクトを精練して再利用可能な部品を開発するクラスデベロッパのそれぞれの手順、およびプロジェクト管理用の手順を提供する。

け標準手順は、要求仕様定義を開発プロセスの対象に含め、要求仕様定義から分析への移行手順を明確に規定している。これにより、要求仕様定義から分析へ展開する際の漏れが減少し、分析で明らかになった仕様を要求仕様定義へ反映することが容易となる。

#### 3.3 オブジェクト抽出ガイドライン

多くのオブジェクト指向開発方法論では、システムに関する記述から、名詞を抜き出すなどの方法が提示されている。しかし、それらはオブジェクトの候補を多数生成するだけであり、適切なオブジェクトを決定することはできない。また、それらの方法はモデルごとに規定されているが、モデル間の関係に関する規定はあいまいであると考えられる。

そのため、クラスデベロッパとアプリケーションカスタマイザ向け標準手順では、分析や設計で多数のオブジェクト候補から適切なものを絞り込む方法を開発した。つまり、オブジェクトの識別、オブジェクトの詳細化、およびオブジェクト間の関連づけという三つの観点を導入し、各観点ごとに、特定の着目点、不要なオブジェクトの排除基準、オブジェクト抽出漏れのチェック基準を設けた。各観点は、モデル横断的なものとした。これにより、オブジェクトの抽出が効果的に行える。

#### 3.4 マルチコンポーネントアーキテクチャ

オブジェクト分析での落とし穴は、担当者が自分のわかっている範囲だけに注力してしまう(時には、わかっているところだけオブジェクトの実装方式まで考えてしまう)ことにより、分析が全体としてバランスよく行われないう点にあると考える。

そのため、クラスデベロッパとアプリケーションカス

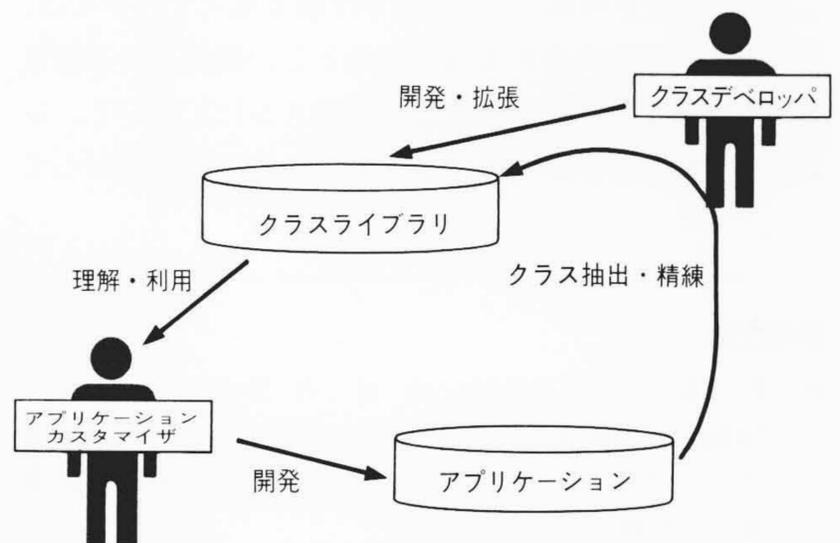


図3 再利用を促進する役割分担

クラスデベロッパとアプリケーションカスタマイザの連携により、品質の良いクラスを部品として再利用することが可能になる。

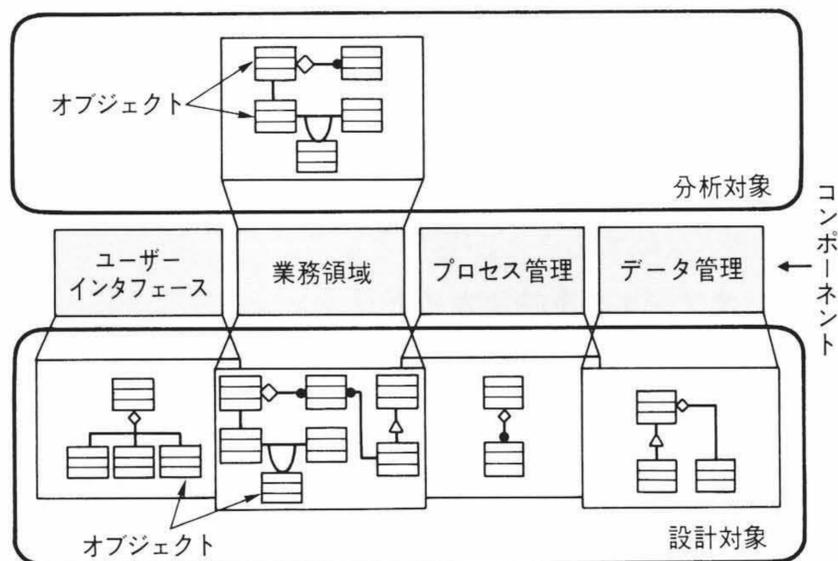


図4 マルチコンポーネントアーキテクチャ

マルチコンポーネントアーキテクチャでは、システムを業務領域、ユーザーインタフェース、プロセス管理、データ管理の四つのコンポーネントで構成する。

タマイザ向け標準手順で、マルチコンポーネントの概念<sup>4)</sup>を導入した。この概念では、システムは業務領域、ユーザーインタフェース、データ管理、およびプロセス管理の四つのコンポーネントで構成する(図4参照)。分析では業務領域コンポーネントだけを検討対象とする。設計では、分析の結果である業務領域コンポーネントに対し他の3コンポーネントを組み合わせる。これら3コンポーネントは実装方式に対応づけられるものである。これにより、分析と設計の明確化を図ることが可能となる。

### 3.5 3ステージによるスパイラル型開発プロセス

日立オブジェクト指向開発標準手順では、仕様明確化、基本仕様実現、システム完成という三つの目的に対応させた3ステージで構成するスパイラル型開発プロセスを開発した(図5参照)。仕様明確化ステージでは、顧客要求の早期確定を目的とし、ビジネスプロセスをモデル化し、業務仕様を明確にする。基本仕様実現ステージでは、システムの骨格部分の実現を目的とし、開発上の不確定要因をプロトタイピングによって明らかにしていく。システム完成ステージでは、システムの骨格部分に対して

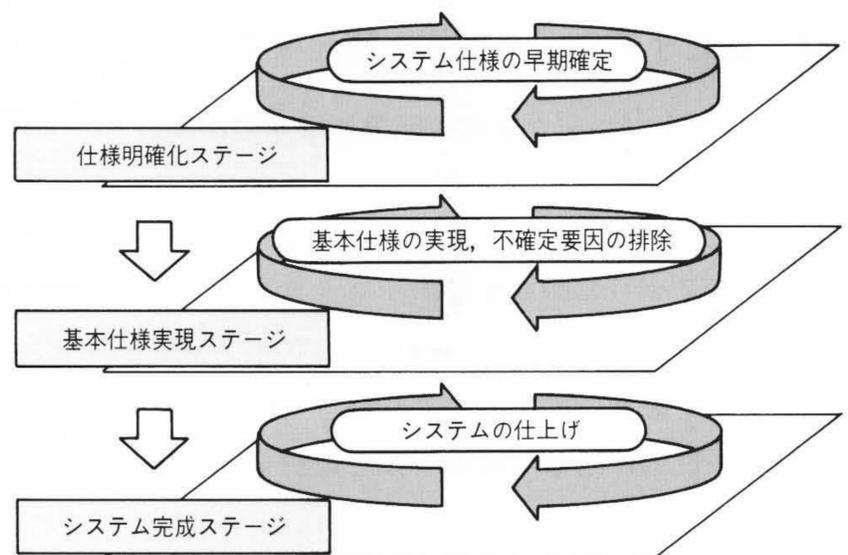


図5 三つのステージによるスパイラル型開発

オブジェクト指向開発標準手順では、目的別に三つのステージを設定し、スパイラル型開発プロセスを効率よく実施する。

すべての機能を作り込む。各ステージでは、「ステップ1」から「ステップ4」までを目的水準に達するまで繰り返す。

このステージ化により、不要な検討を防ぎ、プロジェクト全体の管理を円滑に進めることができる。

## 4 おわりに

ここでは、日立オブジェクト指向開発標準手順の特徴について述べた。この特集では、具体的なオブジェクト指向の開発事例を紹介している。それらは、オブジェクト指向開発標準手順を適用している。

また、従来提供してきたシステム開発方法論HIPACE (Hitachi High-Pace)では、この標準手順を1コンポーネントと位置づけ、ユーザーニーズや開発目的に応じて適切な開発方法を選択することを可能にしている。

今後も、実際の開発時のデータを蓄積し、見積もり精度の向上、クラスライブラリの充実をさらに進め、高品質なシステムをより確実に開発できる技術を提供していく考えである。

## 参考文献

- 1) B. Henderson-Sellers, et al.: A Book of Object-Oriented Knowledge, Prentice Hall(1992) [邦訳: オブジェクト指向ソフトウェア工学—分析・設計・実装—, 大森訳, 海文堂]
- 2) 本位田, 外: オブジェクト指向分析・設計—開発現場に見る実践の秘訣—, 共立出版(1995)
- 3) B. Boehm: Software Engineering Economics, Prentice Hall(1981)
- 4) P.Coad, et al.: Object-Oriented Design, Prentice Hall(1991) [邦訳: オブジェクト指向設計「OOD」, 小畑外訳, プレンティスホール・トッパン]