

㈱NTTデータ 技術開発本部
 明関 恵美 (mda-info-ml@rd.nttdata.co.jp)

1 はじめに

本連載において、前回までにUML2.0の特徴、及びUML1.Xからの変更点について解説を行った。今回は、より具体的な例題を用いて、UML2.0によるモデル記述を行い、システム開発におけるUML2.0ダイアグラムの利用方法の一例を紹介する。

UML2.0は、UML1.Xに比べて厳密にダイアグラムを記述できるという特徴がある。今回はその特徴に注目し、UML2.0とUML1.Xのモデル記述の相違点を中心に解説を行う。

なお、本稿はUML1.Xをご存知の読者を対象としたものである。UML1.Xの詳細に興味がある読者は、恐縮だが、本誌で過去に連載されたUML 紹介記事^[1]等を参照していただきたい。

2 解説に用いるダイアグラム

UML2.0の仕様^[2]で定義されているダイアグラムの一覧を表1に示す。それぞれのダイアグラムの詳細については前回までの連載記事^[3]に記述してあるため、ここでは省略させてい

ただく。

ここでは、実際のシステム開発におけるUML2.0のダイアグラム利用方法の一例を示すことが目的であるため、代表的なダイアグラムのみを用いてモデル記述を行うこととする。

静的構造を表すダイアグラムからはクラス図、コンポジット構造図、コンポーネント図をとりあげる。また、振る舞い図からは、シーケンス図、相互作用概要図、ユースケース図、アクティビティ図をとりあげる。

3 解説で用いる開発プロセス

ここで、UML2.0のモデル記述の際に利用する開発プロセスについて説明する。

UMLは表記法であり、開発プロセスは特定されない。そこで、今回はUnified Process(以下UP)^[4]を用いることとする。UPは、ワークフローを反復的に実施し、全プロセスをフェーズという単位で管理する開発プロセスである。ワークフロ

表1 UML2.0のダイアグラム一覧

分類	ダイアグラム名	
構造図		クラス図
		パッケージ図
		オブジェクト図
		コンポジット構造図
		コンポーネント図
		配置図
振る舞い図	相互作用図	シーケンス図
		コミュニケーション図
		相互作用概要図
		タイミング図
		ユースケース図
		状態機械図
		アクティビティ図

ーとして「要求」「分析」「設計」「実装」等が定義されている。今回は、解説を簡単にするため反復やフェーズは考慮せず、ワークフロー「要求」と「分析」に絞ってモデル記述を行う。また、成果物については、UMLダイアグラムの記述例を紹介することが目的であるため、最小限のUMLモデルのみを作成対象とする。

それぞれのワークフローでは次のように作業を実施する。要求ワークフローではシステムの持つ機能に着目し、「システムへの要求」を定義する。

分析ワークフローでは「どんなシステムを作るか」に注目してオブジェクトの抽出を行う。その過程で、オブジェクト同士の関係をあらゆる構造の分析や、オブジェクトの相互

作用について分析する。

各ワークフローの成果物となるUMLダイアグラムは以下の通りとする。

【要求ワークフロー】

- ・ユースケース図

【分析ワークフロー】

- ・分析クラス図（クラス図）
- ・シーケンス図
- ・アクティビティ図

4 解説で用いる業務例題

では、次に例題となる業務について説明する。例題業務はWebサイト上のパソコン販売であり、以下のような動作を行う。

- ・購入者はパソコン販売システムにアクセスしてパソコンの検索を行

い、パソコンの購入を行う。

- ・パソコン検索結果一覧では、パソコンとその構成部品が一覧で表示される。
- ・パソコン販売システムでは、クレジットカードで決済を受け付け、決済後に商品配送を手配する。

5 UML 2.0 での開発例

ここから、3節で示した開発プロセスに沿って作成した、4節に示した例題業務のUML 2.0のモデルについて解説していく。

5.1 要求

[ユースケース図]

システムの機能をユースケース図で示す(図1)。

UML 2.0のユースケース図では、アクターのアイコンとして人型の他に任意のアイコンを利用できる。図1では、クレジットカード会社のアイコンとしてコンピュータを使うことで、アクターが人でないことが一目でわかるようになっている。

UML 1.Xと比べて異なる主な点は、アクターのアイコンを自由に選べることで、アクターとユースケースの多重度を記述できること等である。

5.2 分析

[分析クラス図]

ユースケース図を元に、クラスを抽出し、クラス構造や関連等を整理

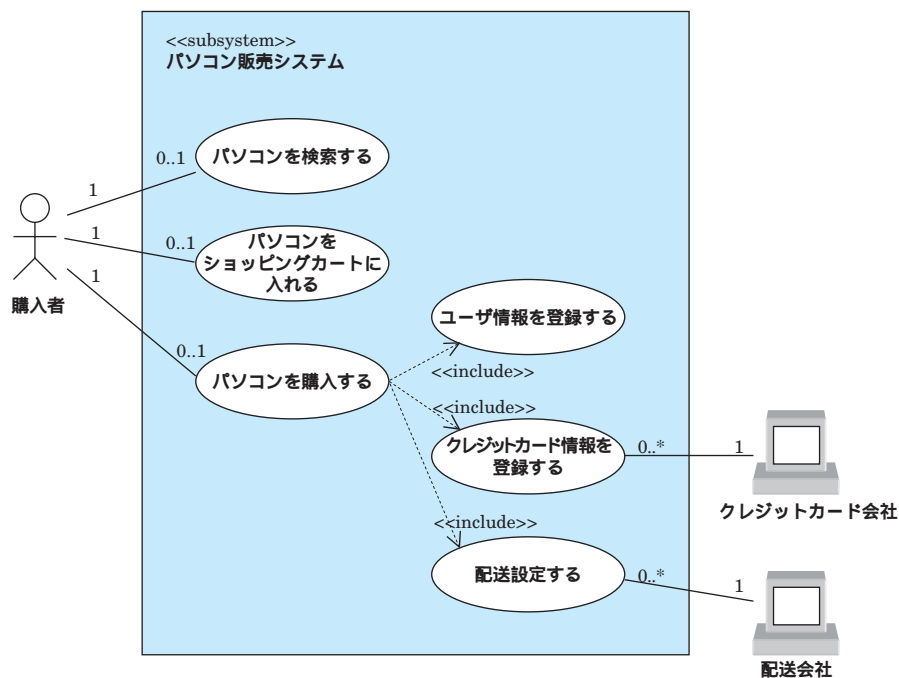


図1 ユースケース図

したUML2.0のクラス図（分析クラス図）の例を図2に示す。

図2では、「パソコン」クラスがコンポジット構造図で表されており、パソコン内部が、「CPU」クラスや「メモリー」クラスといった部品で構成されていることがわかる。また、「CPU」クラスや「メモリー」クラスは「マザーボード」クラスと関連を持っていることも表されている。

UML1.Xの記法では、「パソコン」と部品の関係は、合成集約で記述することができる。しかし、この場合、部品が「パソコン」を構成する要素であることを視覚的に把握することは難しい場合がある。コンポジット構造図で記述することにより、従来の記述方法よりも把握しやすくなっている。

また、図2において、「配送シス

テム」クラスは、「配送手配」インタフェースとして外部に公開しているコンポーネントであることがわかる。「ユーザコントロール」クラスは「配送システム」コンポーネントが公開している「配送手配」インタフェースを利用する、ということも表現されている。

UML1.Xにおいては、インタフェースを利用する側に対する特別な記法は用意されておらず、汎用的に用いられる依存線等を使って表現していた。UML2.0ではインタフェースを要求する側にも記法を与え、単純なアイコンを用いることで、視覚的なわかりやすさを向上させることができた。インタフェースを提供する側として、クラスだけでなくコンポーネントも定義できるようになった。

また、UML2.0において、「認証

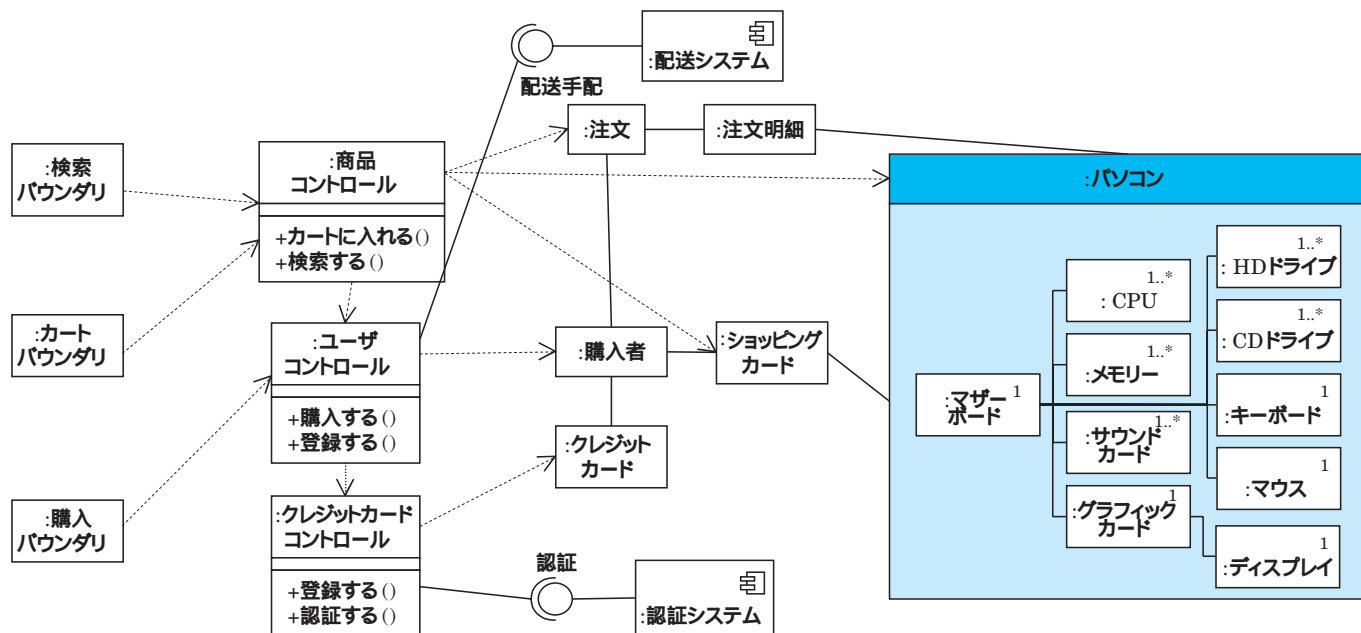


図2 分析クラス図

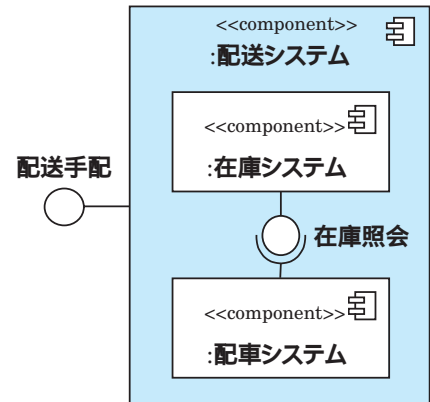


図3 コンポーネントの内部構造を表すコンポジット構造図

システム」、「配送システム」のように複数のクラスから構成されるクラス群を表すコンポーネントは、コンポジット構造図を用いてその内部構造を表すことができる。図3に、配送システムが、在庫システム及び配車システムで構成されている様子を示す。

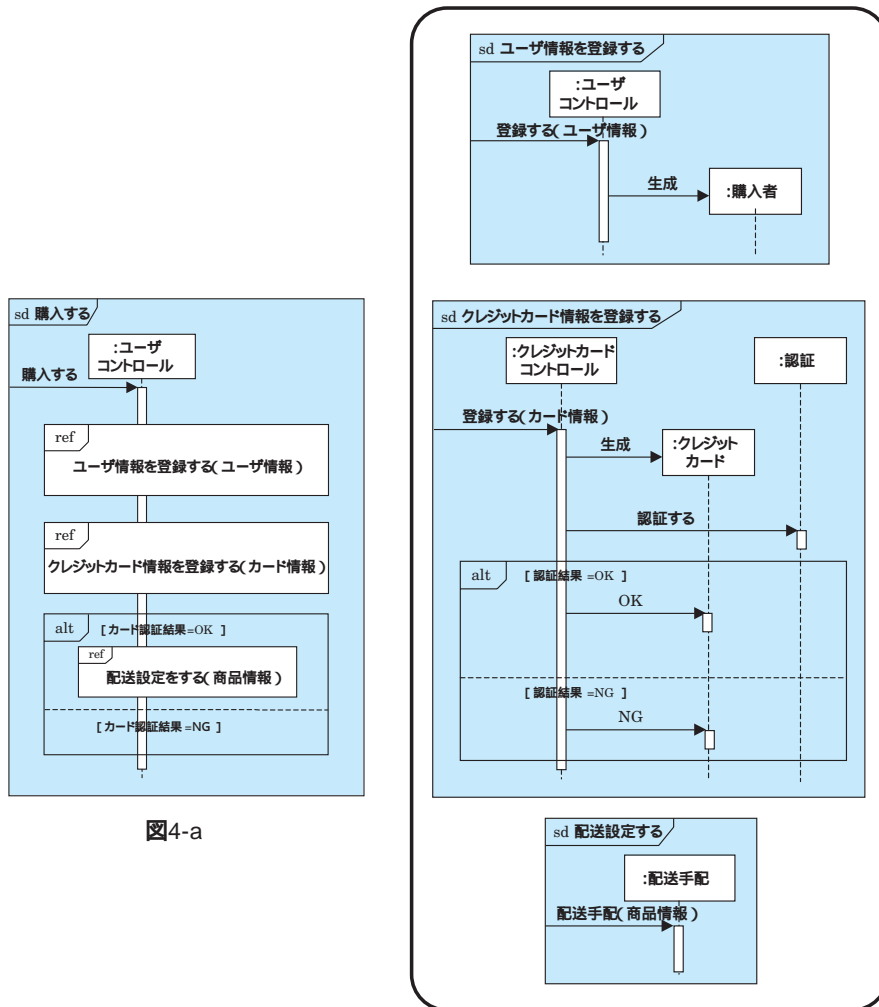


図4-a

図4-b

図4 シーケンス図

[シーケンス図]

図4に、図1におけるユースケース「パソコンを購入する」を実現するオブジェクトの相互作用を表すUML2.0のシーケンス図を示す。

図4-aの「購入する」シーケンス図では相互作用オカレンスを用いて全体の流れが表現されている。図4-bの3つのシーケンス図が、図4-aの「購入する」シーケンス図から参照されている図である。これら

はそれぞれ、図1において、ユースケース「パソコンを購入する」がincludeしているユースケース「ユーザ情報を登録する」、「クレジットカード情報を登録する」、「配送設定をする」を実現するオブジェクトのシーケンス図である。このように、相互作用オカレンスを用いることで、ユースケースのincludeを忠実に表現するシーケンス図を記述することができる。また、図4-bの「ク

レジットカード情報を登録する」シーケンス図では、結合フラグメントを用いてクレジットカードの認証がOKだった場合とNGだった場合の両方の記述を行っている。つまり、シーケンス図内で条件分岐を記述できるため、1つのシーケンス図内で、相互作用を厳密に記述できる。

相互作用オカレンスや結合フラグメントはUML1.Xにはない仕様である。これらの仕様により、これまで厳密には記述できなかった条件分岐の記述が可能になった。

なお、図4-bに示すシーケンス図の実行順序を表すには、相互作用オカレンス以外に、相互作用概要図を用いることができる(図5)。

図6-aに示すような、ユースケース「パソコンを検索する」を実現するクラス「商品コントロール」のメソッド「検索する」の内部動作を記

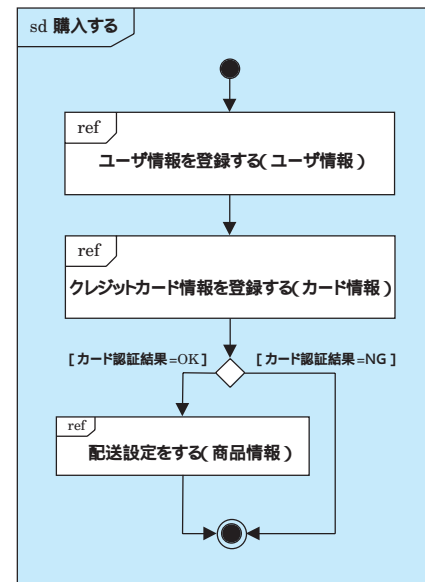


図5 相互作用概要図

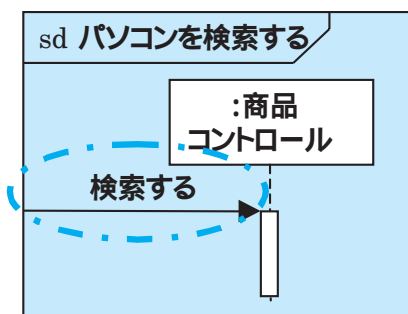


図6-a

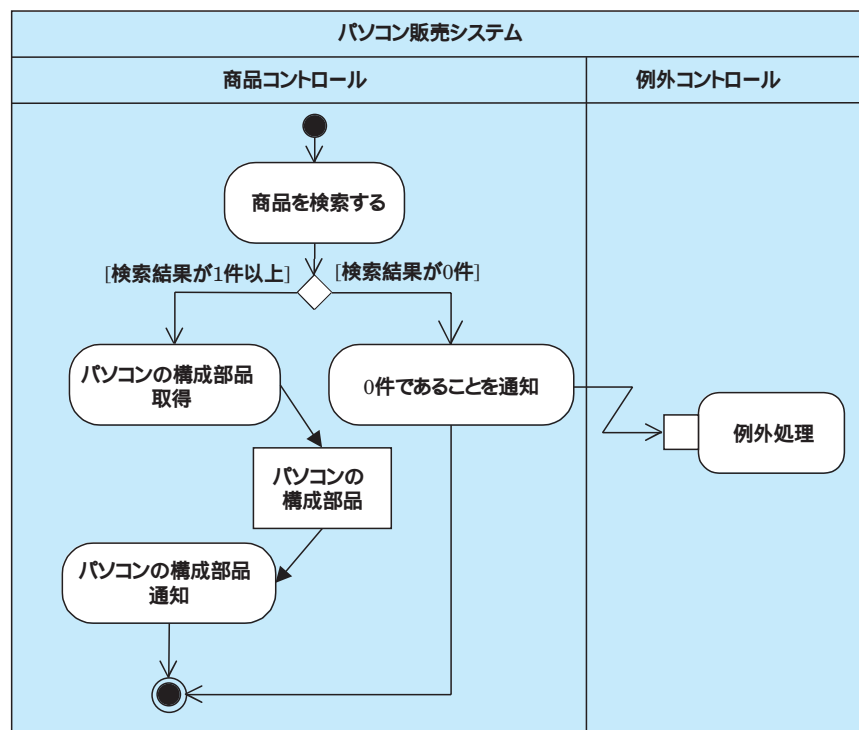


図6-b

図6 メソッド「検索する」の内部処理を表すアクティビティ図

述したUML2.0のアクティビティ図を図6-bに示す。

図では、商品を検索した結果が1件以上だった場合、0件だった場合、例外が発生した場合の動作を表している。検索結果が1件以上だった場合、商品であるパソコンを構成する部品の情報を取得し、呼び出し元に通知している。

図6-bに記述しているように、UML2.0のアクティビティ図はUML1.Xになかった例外処理の表記法が追加されたため、業務概要を表すだけでなく、メソッド内部の動作記述用としても詳細な記述を行うことができるようになった。

6 ダイアグラム間の関係

5.2に示した図1、図4～図6の振る舞い図（ユースケース図、シーケンス図、相互作用概要図、アクティビティ図）の階層関係を図7に示す。

システムの概要を表すユースケース図から、メソッド内の動作を表すアクティビティ図まで、いくつかの振る舞い図を用いて、システム内の動作を概要レベルから詳細レベルまで階層的に表すことができることがわかる。

UML1.Xでは、コラボレーション図とシーケンス図の関係を除き、振る舞い図のダイアグラム間の関係は表現ができなかった。UML2.0でダ

ダイアグラム間の関係が定義でき、大量のモデルを構造化して把握することが可能になった。

7 まとめ

今回は、UML2.0のダイアグラムのうち、主なものを用いたモデル記述の一例を示した。そして、UML2.0のダイアグラムではシステムを厳密に定義できるようになり、ダイアグラム間の関係においては階層構造を定義することができるようになったことを示した。

ここで、UML2.0はUML1.Xに比べてどのような利点があるのかにつ

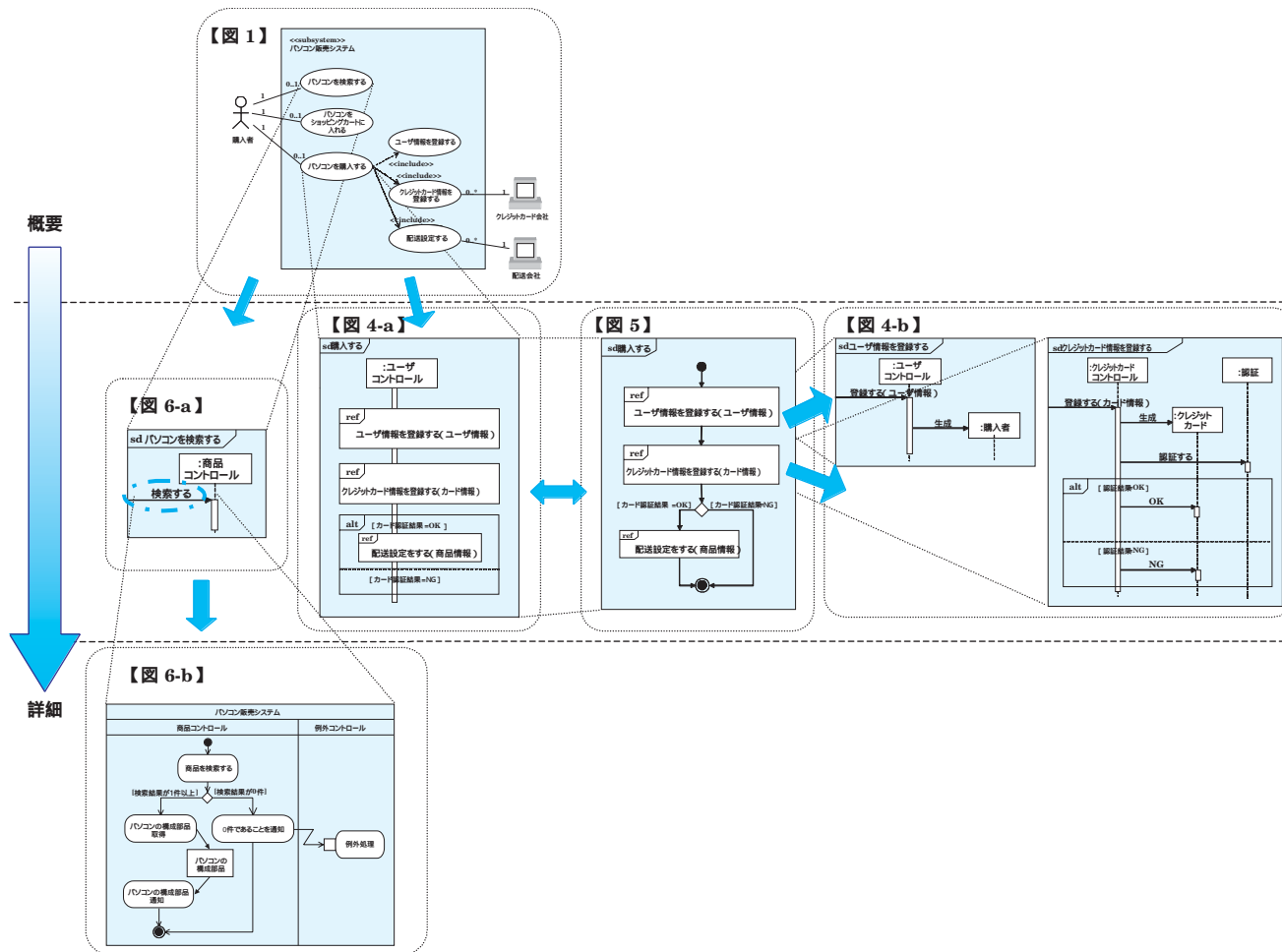


図7 振る舞い図の関係

いてまとめる。

UML2.0は、UML1.Xに比べて厳密な記述が可能である。そのため、概要だけでなく詳細な内容を、図を使って正確に伝えることに適していると言える。また、各ダイアグラム間の関係を明確に定義できるようになったことも重要な利点である。

UML2.0は、これまでに比べて多くのことを厳密に記述できるようになる。しかし、全ての記法を用いて過度な厳密さを求めてモデル記述を

行う必要はない。各開発プロジェクトにおいて、それぞれの作業の目的に応じた記法を選択し、詳細度を決めてモデル記述を行うことが重要である。これはUML1.Xを使う場合と変わらないと言える。

さて、これまで、UML2.0の記法とその記述例を示してきた。最終回である次回は少し視点を変え、UML2.0と今話題になりつつあるMDA (Model Driven Architecture) の関係について解説する予定である。

参考文献

- [1] 山本修一郎, UMLの基礎と応用(連載記事) ビジネスコミュニケーション, Vol.38, No.9, 2001- Vol.40, No.3, 2003
- [2] OMG, UML2.0 Superstructure Final Adopted specification, <http://www.omg.org/UML>, 2003.
- [3] UML2.0入門(本連載)、第1回～第4回、ビジネスコミュニケーション, Vol.41, No.4-7, 2004.
- [4] Ivar Jacobson, Grady Booch and James Rumbaugh, The Unified Software Development Process (1st edition), Addison-Wesley Professional, 1999.