

振る舞い図(1) 相互作用図

(株)NTTデータ 技術開発本部
 滝本 雅之 (mda-info-ml@rd.nttdata.co.jp)

1 はじめに

前回はUML2.0の静的な構造を示す「構造図」について解説した。第3回目の今回は、UML2.0の動的構造（振る舞い図）を表すダイアグラムのうち、相互作用図についてその概要をUML1.Xとの違いを中心に説明する。

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 振る舞い図と相互作用図

UML2.0 Superstructure^[2]（以降UML2.0）では、各ダイアグラムを「システムの静的な構造」に注目した構造図（Structure Diagram）と、「システム内部の挙動（振る舞い）」に注目した振る舞い図（Behavior Diagram）とに大別している（詳細は本連載第1回^[3]を参照のこと）。

今回説明する相互作用図（Interaction Diagram）とは、この振る舞い図の一種でオブジェクト間の呼び出し関係に焦点をあてたダイ

表1 相互作用図

図の名称	レベル	識別名	概要	新規/既存
相互作用図 Interaction Diagram	インスタンス	interaction	オブジェクト間の協調動作を表す	-
シーケンス図 Sequence Diagram	インスタンス	interaction /sd	生存線間のメッセージ交換を時系列で表現する	既存
コミュニケーション図 Communication Diagram	インスタンス	interaction /sd	生存線間のメッセージ交換をその関係にフォーカスして表現する	既存 (改称)
相互作用概要図 Interaction Overview Diagram	インスタンス	interaction /sd	相互作用図間の順序を表現する	新規
タイミング図 Timing Diagram	インスタンス	interaction /sd	時間軸上での状態変化を表す	新規

アグラムの総称である。

UML2.0では、相互作用図として4つのダイアグラムを規定している。表1に示すように、相互作用図は主にインスタンスレベル（UMLのM0レベル相当）の挙動を表現するために作成されるダイアグラムである。

(1) UML1.Xにおける相互作用図の問題点

まず、UML1.Xにおける相互作用図の主な問題点を以下にあげる。

- ・条件により複数の分岐が起こる場合、各状態のスナップショットを別のダイアグラムとして記述しなければならなかった。このため複雑な相互作用がある場合は、「似て非なる」ダイアグラムを大量に記述する必要があった。
- ・オブジェクト数が増加すると、ダ

イアグラムも巨大になる。これが可読性の低下を生み、協調動作の把握をより困難にしていた。

その他にも、実行タイミングを記述する記法がないなど、多くの問題点が利用者より指摘されていた。

次項では、これらの問題点に対する改善策として、UML2.0で追加されたフレームについて説明する。

(2) フレーム

UML2.0の相互作用図では、フレーム（Frame）という新たな記法が追加された。このフレームの導入により、ダイアグラムの階層化や複雑なロジックの記述が可能となった。

フレームは、図1に示すように矩形で表され、左肩の区切った部分に識別名と（必要に応じて）名称を記

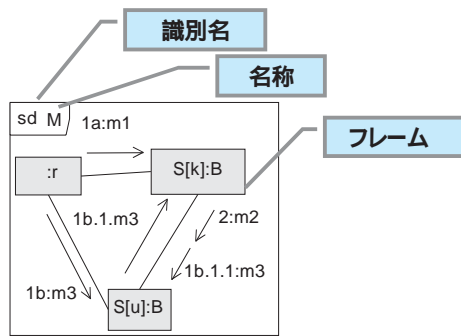


図1 フレーム(コミュニケーション図)

述する。フレームは、識別名により、相互作用 (Interaction) や相互作用オカレンス (Interaction Occurrence)、結合フラグメント (Combined Fragment) を表す。

相互作用は、コンポーネントやクラス、オブジェクト間のメッセージ交換により実現する処理の単位を表す (相互作用図とは異なる定義であることに注意)。相互作用の識別名は、“interaction” または “sd” であり、各図の名称は、識別名の後に記述する (図1)。

相互作用オカレンスは、相互作用の参照を表す。識別名は “ref” であり、参照先名称は矩形の中央に記述する (図2)。

結合フラグメントは、複雑な相互作用を簡素に記述するためのフラグを表す。具体的に規定されている識別名を表2に示す。

次に、図3のシーケンス図を例としてフレームを使った表現を説明する。この例では、以下のことを表している。

- ・処理名はSampleである(相互作用)。
- ・別途定義されたCommonという処理を参照している(相互作用オ

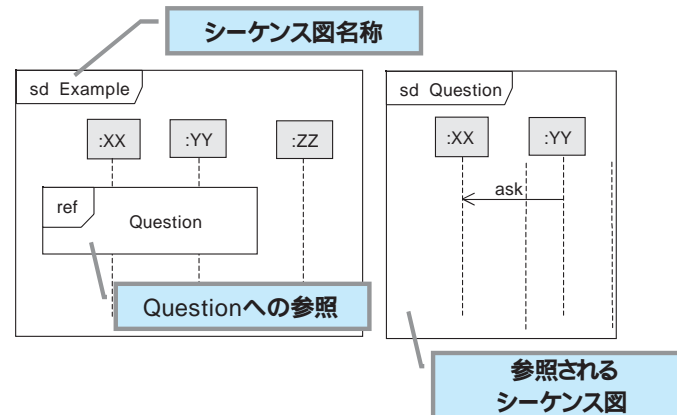


図2 相互作用オカレンスの例(シーケンス図)

表2 結合フラグメント

識別名	説明	識別名	説明
alt	代替フローを表す。つまり if ... then ... else ... のパターン。	ignore	実行時に無視するメッセージを指定する。
loop	そのガード条件が真である間繰り返し処理する。	consider	実行時に考慮するメッセージを表す。
opt	実行するかを判断する。つまり if ... のパターン。	seq	相互作用に弱い順序性があることを表す。
par	並行実行を表す。	strict	相互作用に強い順序性があることを表す。
critical	その領域とトレースしている期間は他からの介入を禁止していることを表す。	neg	この領域へのトレース自体が問題であることを表す。
break	一つ外側の結合フラグメントを中止することを表す。	assert	ガード条件が真の場合、フレーム内部の処理が実行されることを表す。

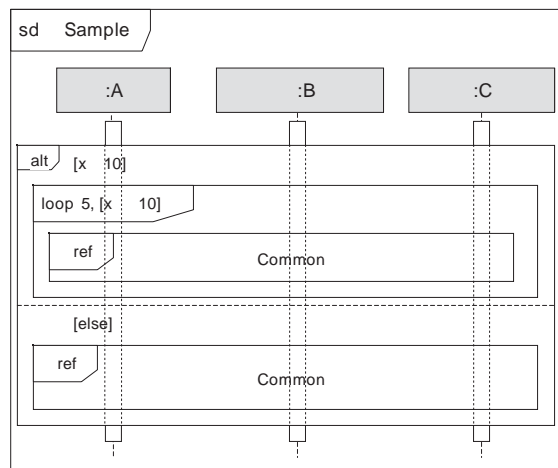


図3 フレームを用いた表記例(シーケンス図)

- ・条件 $x < 10$ の時には、Common の処理を $x < 10$ である限り、最高5回実施する (結合フラグメント)。
- ・ $x < 10$ 以外の場合には、Common を1度だけ実行する。

以上のような複雑な処理であっても、UML2.0では、1つのダイアグラムで表現できるようになった。

(3) 各ダイアグラムの概要

次に、各ダイアグラム概要と変更された背景について、簡単に述べる。

シーケンス図 (Sequence Diagram) は、オブジェクトやクラス、コンポーネント間のメッセージの送受信を時系列に記述するためのダイアグラムである。この図は本連載第2回^[3]で説明したコンポーネント図と同様、記法が大幅に拡張され、その表現能力が飛躍的に向上している。

フレームの追加によって、複雑なロジックの記述や階層化の概念をサポートしたほか、時間制約の記法の追加など、相互作用を更に厳密に記述できるようになっている。

コミュニケーション図 (Communication Diagram) は、オブジェクトやクラス、コンポーネント間のメッセージの送受信を、要素間の関係に焦点をあて記述するためのダイアグラムである (図4)。

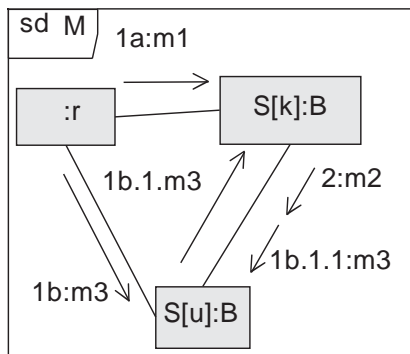


図4 コミュニケーション図の例

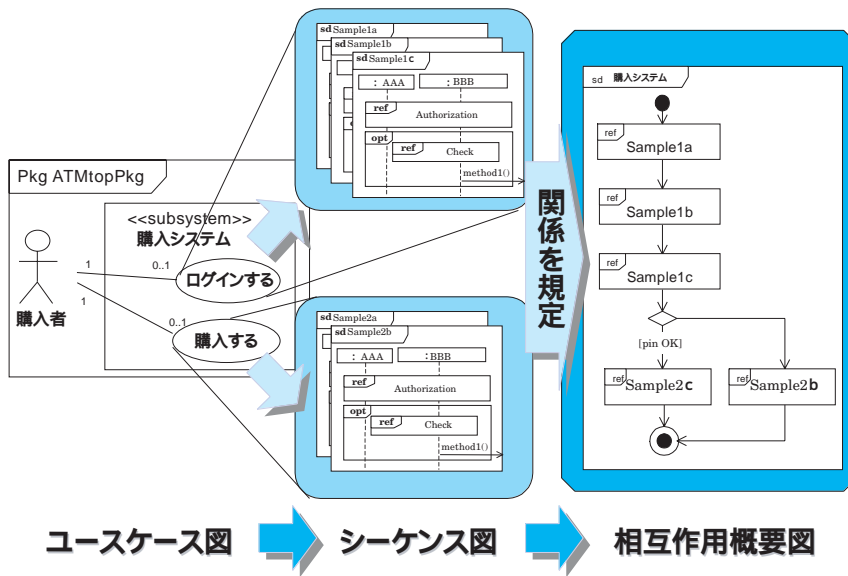


図5 相互作用図の使用例

UML1.Xまではコラボレーション図 (Collaboration Diagram) と呼ばれていたが、クラスやパートの協調動作の構造を表すコラボレーションとの用語の混乱を防ぐ目的で改称された。

この図は、フレームの導入以外、大きな拡張は見送られている。このため今後はシーケンス図の補助的役割を担うことになる見込みである。

相互作用概要図 (Interaction Overview Diagram) は、相互作用図間の順序や分岐を記述するためのダイアグラムである。この図はUML2.0で新規に追加された。このダイアグラムにより、各相互作用図の巨大化を抑制する効果があるとみられる。

タイミング図 (Timing Diagram) は、時間軸に沿って、オブジェクトやクラス、コンポーネントなど各要素における状態の変化とその制約や各要素間の関係を表すダイアグラム

である。この図もUML2.0で新規に追加された。この追加により、各状態と時間や他のオブジェクトとの関係を記述できるようになった。

次節以降は、大幅な拡張がなされたシーケンス図と、新規のダイアグラムである、タイミング図に絞り主な改訂部分を細かく見ていく。なお、相互作用概要図も新規のダイアグラムであるが、その記法は次回解説予定のアクティビティ図と共通な部分が多いため、今回は割愛する。

また、全ての変更点を把握したい読者は、UML2.0仕様書内の、“Change for UML 1.X” という項目を参照していただきたい。

3 シーケンス図

本節では、シーケンス図の変更点について、メッセージに関するものと制約に関するものについて説明する。

(1) メッセージに関する変更点

オブジェクト間のメッセージに関する記法は、大幅に拡張されている。以下では新規に追加されたいくつかの要素について説明する。

メッセージの直接的な送受信はないが、その順序性が重要な場合がある。これを記述するため記法としてジェネラルオーダリング (General Ordering) が用意された。破線内に三角形の矢印を配置し表現する。図6ではジェネラルオーダリングの例をあげている。

また、相互作用オカレンス、つまり階層化をサポートしたことにより、外部からのメッセージ入力で処理が開始されたり、逆に処理結果を外部にむけて出力したりする場合が想定される。このような外部とのメッセージの入出力は、フレームへ向けたメッセージとして記述される。その接点はゲート (Gate) と呼ばれる (図7)。

(2) 制約に関する変更点

次に制約に関する変更点を述べる。

UML 1.Xでは、オブジェクトが受信されてから処理を行っている期間のことを、これまで「活性区間 (activation)」と呼んでいた。

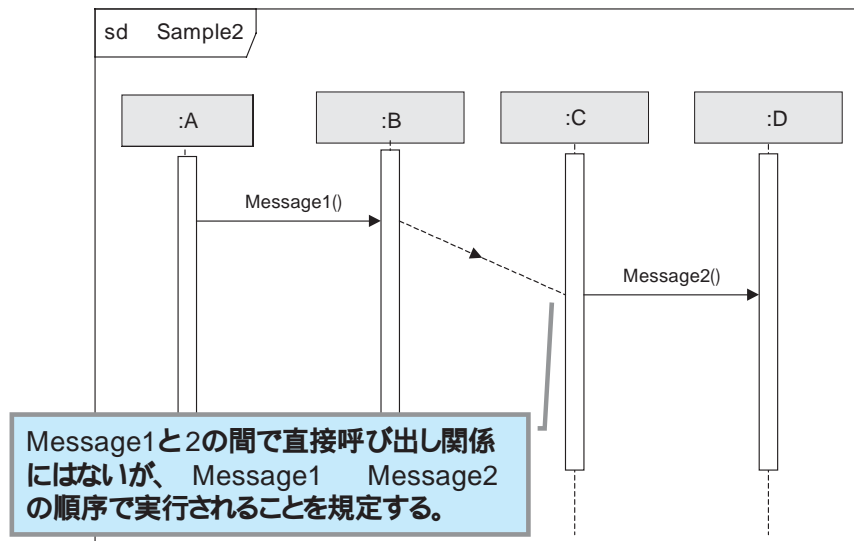


図6 ジェネラルオーダリング

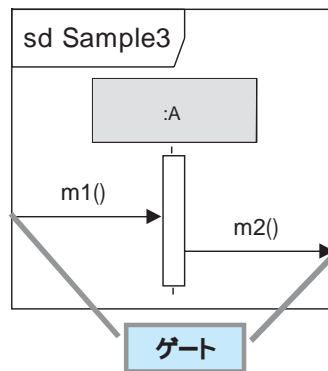


図7 ゲート

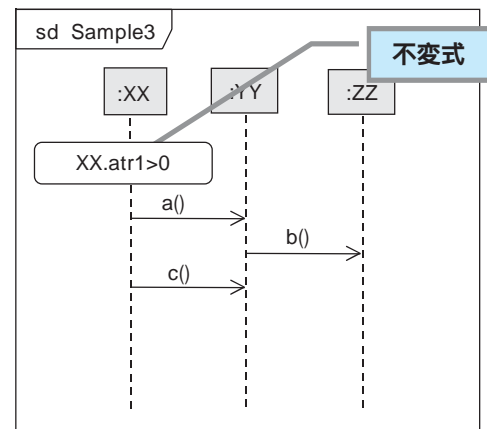


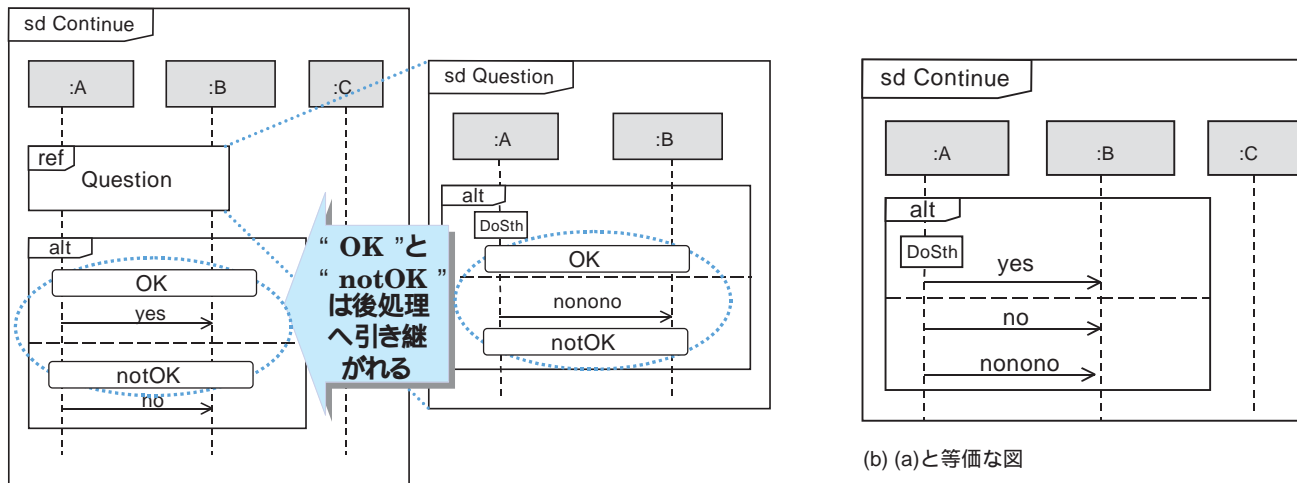
図8 不変式

UML2.0では、これを実行オカレンス (Execution Occurrence) と呼ぶ。

この実行オカレンス上には、状態の不変式 (State Invariant) を記述することができる (図8)。この条件は角丸四角形で表現される。

また、この記法はコンティニューエーション (Continuation) でも利用されている。このコンティニューエーションとは、結合フラグメントの処理中状態を次の処理状態で保持するこ

とを指し、状態名は角丸四角形内に記述する。図9は、コンティニューエーションの例である。この例では、Questionでセットされる“OK”または“notOK”のいずれかの状態が、呼び出し元のContinueに引き継がれている。つまり、Questionで“OK”ならContinueで“yes”というメッセージを送信する。



(a) コンティニュエーションを用いた表現

図9 コンティニュエーション

(b) (a)と等価な図

(3) シーケンス図変更の特徴

シーケンス図において特徴的な変更は、前項の不変式や表2のassertなど、条件や制約に関する記述する記法が多く追加された点である。これはシーケンス図を用いてテストをモデリングすることを視野に入れた拡張と考えられる。すなわち、各クラス間の入出力の値を不変式として厳密に定義することで、結合テストのモデル化が可能となる(図10)。

4 タイミング図

タイミング図はUML2.0で新規に導入されたダイアグラムである。この図は、時間に焦点を当てオブジェクト間の相互作用を表す(図11)。例えば、シーケンス図に対して時間的制約を与える事が可能である。以下では、タイミング図の各構成要素について説明する。

(1) ライフライン

ライフライン(Lifeline)とは相

互作用中にメッセージ交換を行うコンポーネントやクラスやオブジェクトなどのことを指す。

また、単一のライフラインのタイミング図を記述する方法としてジェネラルバリューライフライン(general value lifeline)という記

法がある。この図では上下の線の交点で状態が遷移する(図12)。

(2) タイムライン

状態または条件のタイムライン(State or Condition timeline)は、ライフラインにおける状態を表す線

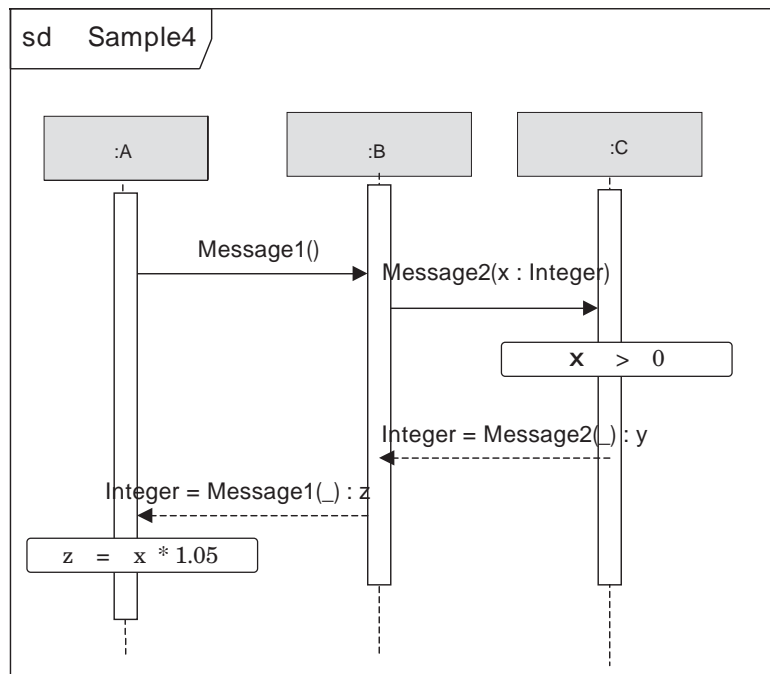


図10 シーケンス図をテストケース及び判定条件の記述例

である。各状態名はライフライン内に記述する。

(3) イベント

イベント (Event) は状態が遷移するきっかけを書く (図11)。イベントはメッセージを受信することで起動する。

(4) タイミングルーラとティックマーク値

タイミングルーラ (Timing Ruler) は目盛り、ティックマーク値 (Tick Mark Value) は、目盛りの値を示す。

(5) 時間観測と持続観測

時間観測 (Time Observation) と持続観測 (Duration Observation) は、時間の変数の定義と時間間隔を示す。例えば図11において、dは時間間隔を、t=nowという式におけるtはロック解除が起きた時刻を表す。

(6) 時間制約と持続制約

時間制約 (Time Constraint) と持続制約 (Duration Constraint) は、その時間が起こる最短・最長時間と、その状態に留まる最低・最長期間を表す。

タイミング図は、基本的に組み込み系モデリングのために追加された記法である。この概念はこれまでのUMLに欠落していた。

ただ、シーケンス図などでも同様の時間的概念が導入されているため、実開発上で今後どの程度利用さ

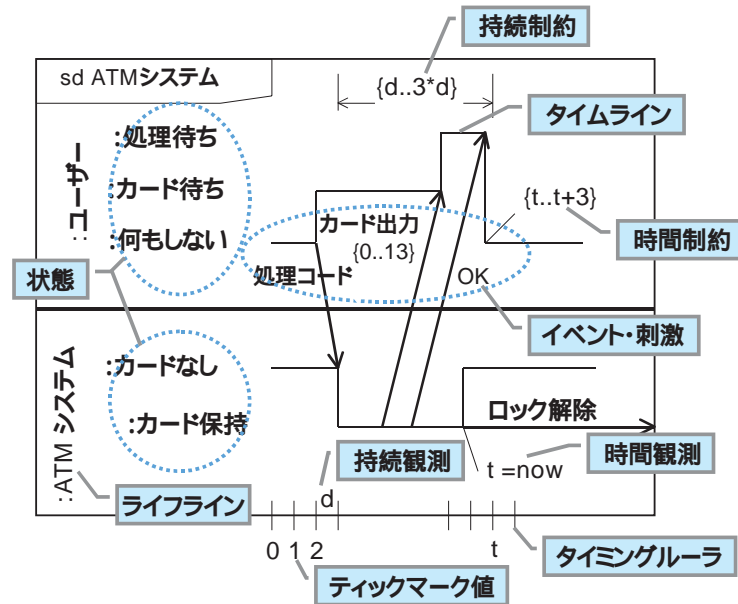


図11 タイミング図の例

れるかは、ツールのサポート次第と言えよう。

5 まとめ

今回は、相互作用図について説明した。相互作用図では主に以下のような記法が追加された。

- ・階層化の記法
- ・複雑な相互作用の記法
- ・時間制約の記法

UML2.0の相互作用図を利用することで、より厳密な相互作用を記述できるようになる。また、ダイアグラムのスケーラビリティの向上も期待できる。

今回は振る舞い図の2回目としてアクティビティ図 (Activity Diagram) と状態機械図 (State Machine Diagram)、ユースケース図 (Use Case Diagram) について解説

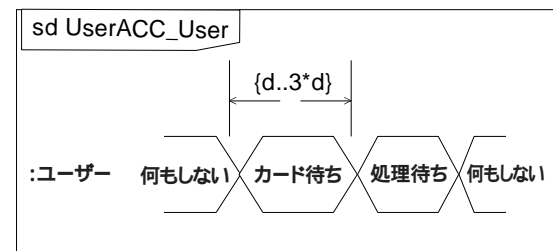


図12 ジェネラルバリューライフラインを用いた例

する。更にアクション意味論 (Action Semantics) とOCL (Object Constraint Language) についても触れる予定である。

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2回, Vol.41, No.4-5, 2003