

第4 & 5講 UMLの概要(1)

- UMLとは
- オブジェクト図
- クラス図

モデリングとモデル

- モデリング
 - 実世界の事柄を別の物体で表現すること
 - モデルを作成すること
 - プログラミング
 - 処理をプログラム言語という手段で表現
 - ↓
 - オブジェクト指向
 - データ構造をオブジェクトの属性
 - 処理を振る舞いとしてモデリング
- モデル
 - ある視点から見たシステムの抽象的な表現
 - ダイアグラム(図)により表現
- モデリング言語
 - モデルを表現する手段

UMLとは

- UMLとは
 - Unified Modeling Language (統一モデリング言語)
 - ソフトウェア主体のシステムの成果物をビジュアル化、仕様化、構築、文書化するためのグラフィカルな言語
 - オブジェクト指向分析・設計を表現するためのモデリング言語(モデル表記法を統一したもの)
 - モデリング言語の国際標準(OMG標準、ISOでの標準化が進められている)

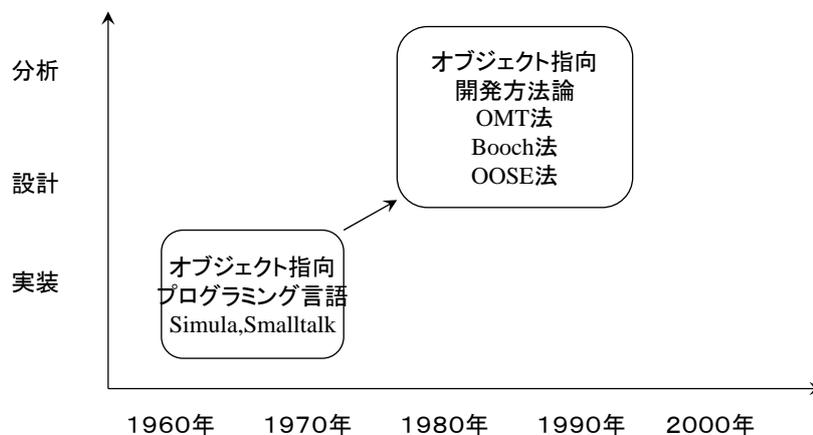
UMLとモデル

- UMLモデルは図で表現
 - 一つの図でモデルを表現できない。様々な観点(ビュー)から複数の図を記述することが必要。
- UMLは図を使って様々な要素を表現するために、それぞれの要素には図での書き方が決まっている。この書き方を表記法(Notation)という。たとえば、インターフェイスには、2つの表記法が存在する。

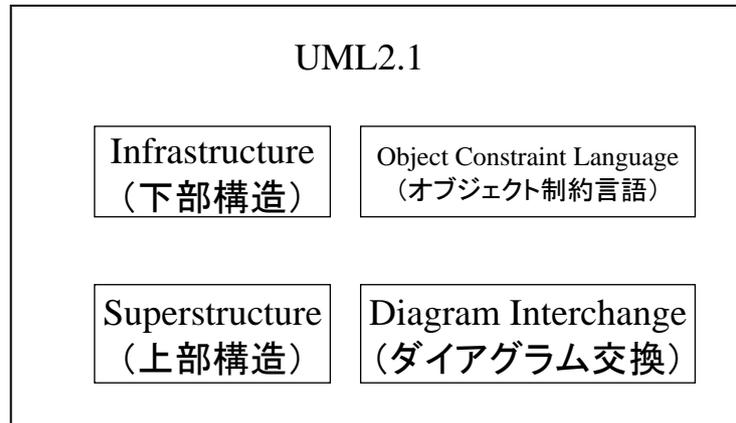
UMLと開発モデル

- UML
 - オブジェクト指向分析/設計に使用するダイアグラム(図)の描き方の文法
- 開発モデル(プロセス)
 - オブジェクト指向分析/設計をおこなうための方法論(システム開発の方法論)

オブジェクト指向開発方法論の流れ



UML2.1の構造



UML1.4のモデル(図)

- クラス図
- ユースケース図
- シーケンス図
- オブジェクト図
- コラボレーション図
- ステートチャート図
- アクティビティ図
- コンポーネント図
- 配置図

UML1.4の主要モデル

- 利用者から見たシステムの機能を表すモデル
(機能モデル)
- 問題領域やシステムの静的構造を表すモデル
(静的モデル)
- 振る舞いを表す動的モデル(動的モデル)
- 構成モデル(物理的なモデル)

UML1.4の9つのダイアグラム(1)

- 利用者から見たシステムの機能を表すモデル
(機能モデル)
 - ユースケース図
 - アクティビティ図
- 問題領域やシステムの静的構造を表すモデル
(静的モデル)
 - クラス図
 - オブジェクト図

UML1.4の9つのダイアグラム(2)

- 振る舞いを表す動的モデル(動的モデル)
 - ステートチャート図(状態遷移図)
 - シーケンス図
 - コラボレーション図(協調図)
- 構成モデル(物理的なモデル)
 - コンポーネント図
 - デプロイメント図(配置図)

UML2.1の図

- UML2.1のSuperStructureで定義されている13種類の図
 - 構造に注目してモデリングするための構造図
 - 振る舞いに注目してモデリングするための振る舞い図

UML2.1の構造図(1)

- モデリングする対象の構造着目にしてモデリングするための図
 - クラス図
 - オブジェクト図
 - パッケージ図
 - コンポジット図
 - コンポーネント図
 - 配置図

UML2.1の振る舞い図(2)

- モデリングする対象の振る舞いにしてモデリングするための図
 - ユースケース図
 - アクティビティ図
 - 状態マシン図
 - 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

オブジェクト図 (1)

- クラス図
 - クラスの構造やクラス間の関連をあらわすためのダイアグラム
- オブジェクト図
 - クラス図に描かれたクラス群の実体のある時点における様子表現したダイアグラム
 - クラスシンボルのインスタンス形であるオブジェクトシンボルを用いて表現

オブジェクト図 (2)

- オブジェクト
 - クラスの実体であり、インスタンスとも呼ばれる
- オブジェクトの表現
 - オブジェクトは長方形で表現される
 - 長方形の中には、オブジェクト名:クラス名の形式で記述される
 - クラス名やオブジェクト名は省略可能
 - クラス名だけを記述する場合には、“:”から記述する
 - オブジェクトの長方形を2つの区画に分けて、属性を表現できる
 - 属性は、属性名:型=値の形式で表す

オブジェクト図 (3)

- リンク
 - クラスに対して実体(インスタンス)があるように、関連の実体をリンクという
 - 関連と同様に、実線で表す
 - オブジェクト同士に何らかの関係があることを表す
 - リンクはインスタンスなので、多重度を定義できない。すなわち、リンクの多重度は常に1である。
 - リンクの端には、オブジェクトの役割を記述する

クラス図

- 各クラス間の関係を表現することで、システムの静的な構造を表現
- クラス図は、クラス名、属性、メソッド(操作)という3つの部分から構成
- 一般に、長方形で表され、横線で3つの部分に分けられる
 - 属性
 - 「可視性 名前:型=デフォルト値」
 - メソッド
 - 「可視性 名前(引数1, 引数2, …):返り値」

アクセス指定(制御)

- 属性やメソッドの可視性を指定
 - +
 - public (どこからでも可視)
 - #
 - protected (パッケージ内および派生したクラスから可視)
 - -
 - private (クラス内でのみ可視)
 - ~
 - package (パッケージ内のみで可視)

継承

- クラス間の継承関係
 - サブクラスからスーパークラスに向かう白抜きの矢印で表現
- 継承の表現
 - 分割表記
 - 共有表記

継承

- 継承
 - 既存のクラスに属性や操作を追加して、新しいクラスを定義すること
 - 共通の属性と振る舞いを抽象化して、新しいクラスを作成できる
- スーパークラスとサブクラス
 - スーパークラス(親クラス)
 - あるクラスが継承によって定義された場合、その継承元のクラス
 - スーパークラスを継承したクラスをサブクラスという
 - サブクラス(子クラス)
 - あるクラスから、継承によって定義されたクラス
 - 継承元のクラスをスーパークラスという

汎化と特殊化

- 汎化
 - 複数のクラスの共通の特性を抽出してクラスを作ること
 - 汎化してできるクラスをスーパークラスとよぶ
 - 汎化の関係をis-a関係とよぶ
- 特殊化
 - ひとつのクラスを特性の違いに注目して分類すること
 - 分類されたクラスをもとのクラスのサブクラスという
 - 特殊化の関係をa-kind-of関係とよぶ

関連

- クラス間における関連
 - クラス間に結びつきがあることを表す
 - クラス間に直線を引いて表現
 - 関連名は線の近くに記述

多重度

- 関連するクラスのオブジェクト(インスタンス)数を表現
- 一方のクラスのオブジェクトとリンクすることができる
- 多重度の表現
 - * : 0以上
 - 1..* : 1以上
 - 0..1 : 0または1

集約

- クラス間の関係が「部分—全体」であるという関連を表現
- has-aを表現
- 特殊な関連で、関連の「全体」側に白抜きの菱形を付けて表現

コンポジション

- 集約の一種
- 「全体が消滅した場合、部分も消滅する」という強い条件が該当する集約
- 「全体」を表すクラスの端に塗りつぶした菱形を付けて表現

コンポジション

- オブジェクトを他のオブジェクトで構成すること
 - コンピュータ: ビデオカード、キーボード、ディスクドライブなど
 - テレビ: チューナ、ブラウン管など
- コンポジション関係はhas-a関係とよばれる
 - オブジェクトに対して、メッセージを動かすように依頼するものをメッセージと呼ぶ