

ソフトウェア工学

07: UMLモデリング(Ⅱ) – システムの静的側面のモデリング

理工学部 経営システム工学科
庄司 裕子



今回のテーマ

UMLモデリング(Ⅱ) – システムの静的側面のモデリング

- ❖ クラスとオブジェクト
- ❖ クラス図
- ❖ クラス間の関係
- ❖ クラスの操作
- ❖ クラス図の拡張的概念
- ❖ パッケージによるクラス図の組織化

2

UML (Unified Modelling Language) とは

ソフトウェアシステムの成果物を

- ❖ 仕様化 (specify)
- ❖ 視覚化 (visualize)
- ❖ 作成 (construct)
- ❖ 文書化 (document)

するための汎用のビジュアル モデリング言語

3

UMLダイアグラム

- | | |
|--|---|
| <ul style="list-style-type: none"> ❖ <u>システムの静的(構造</u> <u>的)側面</u> ⇒ クラス図 ⇒ オブジェクト図 ⇒ パッケージ図 ❖ コンポーネント図 ❖ 配置図 | <ul style="list-style-type: none"> ❖ <u>システムの動的側面</u> ❖ ユースケース図 ❖ シーケンス図 ❖ コラボレーション図 ❖ アクティビティ図 ❖ ステートチャート図 |
|--|---|

4

UMLモデリング(Ⅱ) – システムの 静的側面のモデリング

- ⇒ クラスとオブジェクト
- ❖ クラス図
- ❖ クラス間の関係
- ❖ クラスの操作
- ❖ クラス図の拡張的概念
- ❖ パッケージによるクラス図の組織化

5

クラスとは

- ❖ 同じ属性、操作、関係、およびセマンティクスを共有するオブジェクトの集合を記述したもの
 - ❖ 属性値はオブジェクトごとに異なる場合がある
- ❖ クラスは抽象概念 (abstraction) である
 - ❖ 共通する特性を抜き出す (抽象)
 - ❖ その他の特性を捨て去る (捨象)
- ⇒ 実世界の存在ではない (設計時の概念)

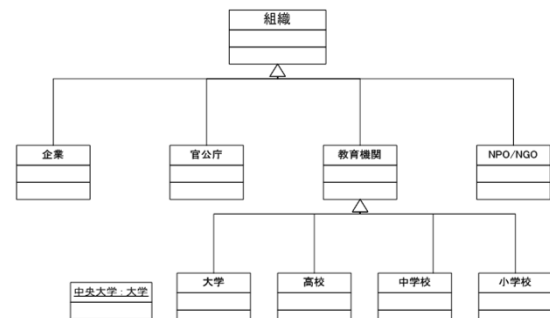
6

オブジェクトとは

- ❖ オブジェクトはクラスのインスタンス(実例)
- ⇒ 実世界に存在する(実行時の概念)
- ❖ 実行時にはクラスは存在せず、オブジェクトのみ
- ❖ UMLダイアグラムでは、シーケンス図、コラボレーション図などに登場する

7

クラスとオブジェクト(例)



8

UMLモデリング(Ⅱ) – システムの静的側面のモデリング

- ✓ クラスとオブジェクト
- ⇒ クラス図
- ❖ クラス間の関係
- ❖ クラスの操作
- ❖ クラス図の拡張的概念
- ❖ パッケージによるクラス図の組織化

9

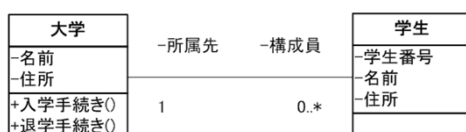
クラス図

- ❖ システムの静的ビューを表現する
 - ❖ クラスの内部構造
 - ❖ クラス間の関係
- ❖ 3つの観点
 - ❖ 概念的観点 → ドメインの概念モデリング
 - ❖ 仕様の観点 → ソフトウェアのインタフェース
 - ❖ 実装の観点 → ソフトウェアの実装
- ⇒ さまざまなUMLダイアグラムの中で最も多用される重要なダイアグラム

10

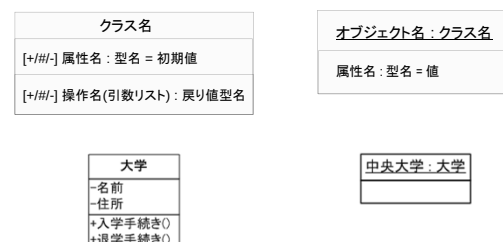
クラス図のUML表記

- ❖ 四角形の中にクラス名を書く
- ❖ 必要に応じて、四角形を3つの区画に分け、それぞれに名前、属性、操作を記述できる
- ❖ クラス間の関連は線で表され、その両端(関連端)にロール名と多重度を明示できる



11

クラス/オブジェクトの表記法と例



12

クラス図の目的

- ❖ 通常、システムの静的ビューをモデリングする際に、次の3つのうちいずれかの目的に使用される
 - ❖ システムの語彙のモデリング
 - ❖ 単純な(クラスレベルの)コラボレーションのモデリング
 - ❖ 論理データベーススキーマのモデリング
- ⇒ これらは3つの観点に対応する

13

オブジェクト図

- ❖ システムに存在するオブジェクトのある時点でのスナップショット
- ❖ 一般に複数のオブジェクトから成る
- ❖ メッセージのないコラボレーション図とみなせる

14

UMLモデリング(Ⅱ) – システムの静的側面のモデリング

- ✓ クラスとオブジェクト
- ✓ クラス図
- ⇒ クラス間の関係
- ❖ クラスの操作
- ❖ クラス図の拡張的概念
- ❖ パッケージによるクラス図の組織化

15

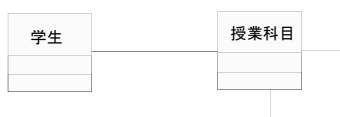
クラス間の関係

- ❖ クラスレベルのコラボレーションを表す
 - ❖ オブジェクト間の関係はコラボレーション図で示される
- ❖ 関係の種類
 - ❖ 関連
 - ❖ 集約、コンポジション(関連の一種)
 - ❖ 汎化

16

関連とは

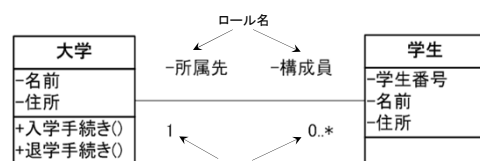
- ❖ 指定された2つ以上の分類子(クラス、アクター、ユースケース、サブシステム、コンポーネントなど)間の意味的な関係で、それらの分類子のインスタンス間の結び付き(connection)を記述するもの
- ❖ UMLでは関連のインスタンスを「リンク」と呼ぶ



17

関連のUML表記

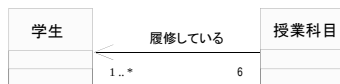
- ❖ クラス間の関連は線で表される
- ❖ 関連線の両端(関連端)にロール名、多重度、誘導可能性などを、関連線に関連名を、それぞれ明示できる(必須ではない)



18

誘導可能性

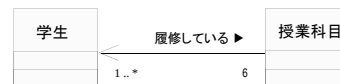
- ❖ 関連を構成する一方のクラスから、関連をたどって相手先のクラスを誘導(ナビゲート)できるかどうかを、関連端の矢印の有無で示す
- ❖ (例)「授業科目」オブジェクトは「学生」オブジェクトを参照できる



19

関連名

- ❖ 関連の意味を明確にするために、関連に名前を付けることができる
- ❖ 名前は関連線の中央のラベルで表される
- ❖ 関連名は一般に動詞または動詞句
- ❖ 黒の三角形で関連の方向を示すことが可能
 - ❖ 「学生」は「授業科目」を「履修している」



20

ロール名

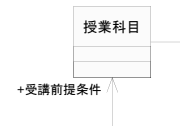
- ❖ 関連の中でクラスがどのような役割を果たすのか(どのような“顔”を見せるのか)を指定する
- ❖ ロール名は一般に名詞または名詞句である
- ❖ ロール名は関連線に沿って、その名前が修飾するクラスの近くに配置される
 - ❖ ロール名は関連の片方または両方の関連端に付けることができる



21

再帰的関連とロール名

- ❖ 再帰的関連では、同じクラスのオブジェクト同士が関係付けられる
- ❖ 同じクラスの複数のオブジェクトが何らかの形で互いに協調することを示す
- ❖ 再帰的関連ではロール名を使う必要がある



22

多重度

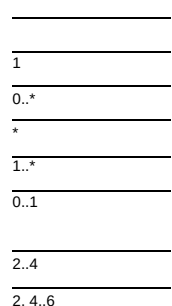
- ❖ 特定のクラスについて、別のクラスの「1つの」インスタンスと関係するインスタンスの数の範囲を示す
- ❖ 各関連の両方の関連端にそれぞれ対応する多重度を指定する
 - ❖ 教授のインスタンスについては、担当するコース科目は1つとは限らない(担当科目がないかもしれないので0以上)
 - ❖ 授業科目のインスタンスについては、講師である教授は0名または1名のどちらかである



23

多重度表記のバリエーション

- ❖ 指定なし
- ❖ 厳密に1
- ❖ 0以上(多数、上限なし)
- ❖ 1以上
- ❖ 0または1(オプションの多重度1)
- ❖ 範囲を指定
- ❖ 複数の範囲を指定



24

多重度の意味するもの

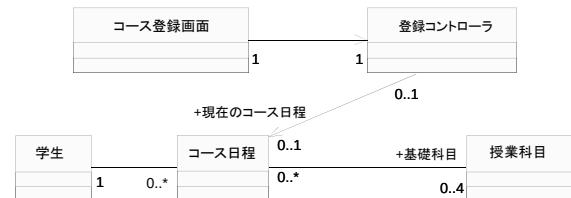
- ❖ 多重度は2つの疑問に答える
 - ❖ 関連は必須か、それとも省略可能か
 - ❖ 特定のインスタンスにリンク可能なインスタンス数の下限および上限は？



25

考えてみよう！

- ❖ コース日程に組み込み可能な授業科目の数は？
- ❖ 各コース日程には何名の学生を割り当て可能か
- ❖ 日程は学生の割り当てなしで存在できるか
- ❖ コース登録画面では何個のコース日程をオープンできるか



26

集約 (aggregation)

- ❖ 関連の一種で、全体とその部分の間に成立する「全体-部分」関係を表すもの
 - ❖ 集約は「の一部 (is a part of)」関係である
 - ❖ 部分側は全体側とは独立に存在できる
- ❖ 通常の関連と同様に、多重度を表現できる
- ❖ 白抜きひし形を全体側の関連端に付けて示す



27

コンポジション (composition)

- ❖ より強い集約関係を表すもの
 - ❖ 部分オブジェクトは唯一そのリンク先の全体オブジェクト内に存在し、全体オブジェクトと共に存続し、共に消滅する
 - ☞ いわば、埋め込み型 (集約は参照型)
- ❖ 黒塗りのひし形を全体側の関連端に付けて示す



28

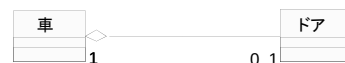
集約の判定条件

- ❖ 関係の記述に「の一部」という語句が使われるか
 - ❖ ドアは車「の一部」である
- ❖ 全体側のいくつかの操作が、その部分に自動的に適用されるか
 - ❖ 車が移動すると、ドアも移動する
- ❖ 全体側のいくつかの属性値が、その部分の全部または一部に伝播されるか
 - ❖ 車が青色なら、ドアも青色である
- ❖ 関係が本質的に非対称的であり、一方のクラスが他方に従属するか
 - ❖ ドアは車の一部であるが、車はドアの一部ではない

29

関連か集約か

- ❖ 2つのオブジェクトが全体-部分関係で強く結び付く場合
 - ❖ 両者の関係は集約である



- ❖ 2つのオブジェクトが通常は互いに独立していると見なされるが、両者がしばしばリンクされる場合
 - ❖ 両者の関係は関連である



30

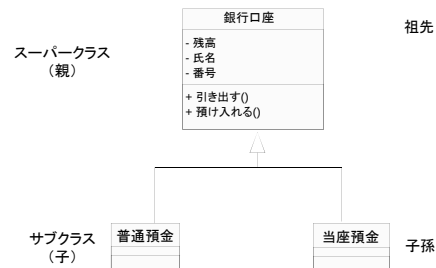
汎化とは

- ❖ クラス間の関係で、1つのクラスが、1つまたは複数のクラスの構造 and/or 振る舞いを共有（継承）することを表す
- ❖ 分類階層関係（「一般-特殊」の関係）を定義する。この階層内では、サブクラスが1つまたは複数のスーパークラスを継承する
 - ❖ 単一継承
 - ❖ 多重継承
- ❖ 「の一種 (is a kind of)」関係である

31

単一継承

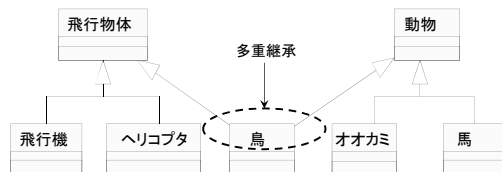
- ❖ あるクラスが別の「1つの」クラスを継承する



32

多重継承

- ❖ 1つのクラスが他の「複数の」クラスから継承する

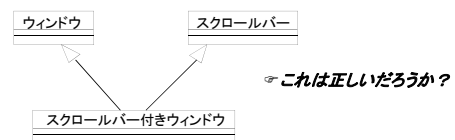


思わぬ結果をもたらすことが多いので、多重継承の使用には常に細心の注意を払い、本当に必要な場合にのみ使用すること

33

汎化と集約を混同してはいけない

- ❖ 汎化と集約はしばしば混同される
 - ❖ 汎化は「である (is-a)」または「の一種 (is-a-kind-of)」関係（分類階層関係）を表現する
 - ❖ 集約は「の一部 (is-a-part-of)」関係（全体-部分関係）を表現する



34

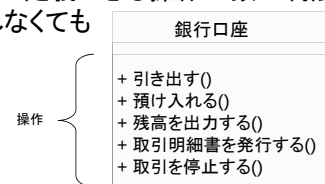
UMLモデリング(Ⅰ) – ユースケースモデリングとユースケース駆動型開発

- ✓ クラスとオブジェクト
- ✓ クラス図
- ✓ クラス間の関係
 - ⇒ クラスの操作
- ❖ クラス図の拡張的概念
- ❖ パッケージによるクラス図の組織化

35

操作(operation)とは

- ❖ クラスのオブジェクトを呼び出して実行できる変更や問い合わせの仕様
 - ❖ それを実装したものがメソッド
- ❖ クラスに定義できる操作の数に制限はなく、定義しなくてもよい



36

操作と責務 (responsibility)

- ❖ クラスとは、クラスに属するオブジェクトの振る舞いを定義する責務の集合を具体的に表現したものである
- ❖ クラスの責務はクラスの実行によって遂行される
 - ❖ 責務と操作の対応は 1 対 1 とは限らない
- ❖ 1 つの操作は、単純でまとまりのある 1 つの機能を実行するのが望ましい

37

操作の命名と記述

- ❖ 適切な操作名を使う
 - ❖ 結果 (得られる機能) を示す名前
 - ❖ 呼び出し側 (クライアント) の視点に基づく名前
 - ❖ クラス間で一貫性を保った名前
- ❖ 操作のシグニチャを定義する
 - ❖ 操作名 (パラメータ: クラス...): 戻り値型
- ❖ すべてのパラメータの意味をまとめた短い説明を付ける

38

UMLモデリング(Ⅱ) – システムの静的側面のモデリング

- ✓ クラスとオブジェクト
- ✓ クラス図
- ✓ クラス間の関係
- ❖ クラスの操作
- ⇒ クラス図の拡張的概念
- ❖ パッケージによるクラス図の組織化

39

クラス図の拡張的概念

- ❖ ステレオタイプ
- ❖ 単一/多重分類と静的/動的分類
- ❖ 派生関連と派生属性
- ❖ 限定子付き関連
- ❖ 関連クラス
- ❖ パラメタライズド クラス
- ❖ その他

40

UMLモデリング(Ⅱ) – システムの静的側面のモデリング

- ✓ クラスとオブジェクト
- ✓ クラス図
- ✓ クラス間の関係
- ❖ クラスの操作
- ❖ クラス図の拡張的概念
- ⇒ パッケージによるクラス図の組織化

41

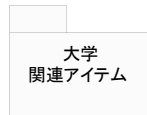
クラス図の組織化

- ❖ 開発対象のシステムに数百個の (場合によっては数千個もの) クラスが含まれることは珍しくない
- ⇒ クラス図を組織化するためのより優れた方法が必要

42

パッケージとは

- ❖ 複数の要素をグループに編成するための汎用のメカニズム
- ❖ 他のモデル要素を含有できるモデル要素
- ❖ 上辺にタブを付けた四角形で表す
- ❖ 使用目的
 - ❖ 開発中のモデル(クラス)を組織化する
 - ❖ 構成管理の単位とする



43

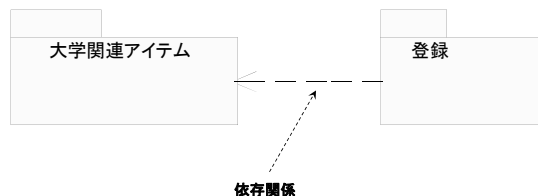
パッケージによる要素の組織化

- ❖ 機能的に関連する複数のモデル要素を1つのパッケージとしてグループ化する
- ❖ 機能的に関連するかどうかの判断
 - ❖ あるクラスへの変更が別のクラスに波及するか
 - ❖ クラスがシステムから「削除された」場合に、他のクラスが深刻な影響を受けるか
 - ❖ クラスが多数のメッセージと相互作用するか

44

例: パッケージを使用したクラス図

- ❖ 「登録」パッケージ内の要素が「大学関連アイテム」パッケージ内の要素に依存している



45

まとめ: UMLモデリング(静的側面)

- ❖ クラスとオブジェクト
- ❖ クラス図
 - ❖ システムの構造を規定
 - ❖ システム開発の際最もよく利用される
- ❖ オブジェクト図
 - ❖ 実行時のある時点に存在するオブジェクトを示すスナップショット
- ❖ クラス間の関係とその記述
 - ❖ 関連、集約、汎化など
- ❖ クラスの操作
 - ❖ そのクラスのオブジェクトがおこなえる処理(→メソッド)
- ❖ パッケージによるクラス図の組織化
 - ❖ 多数のクラスを機能に基づいてグループ化

46

参考文献

- ❖ Perdita Stevens with Rob Pooley, 「Using UML: Software Engineering with Objects and Components, Updated Edition」, Pearson Education (邦訳: 「オブジェクト指向とコンポーネントによるソフトウェア工学 - UMLを使って -」ピアソン・エデュケーション)
- ❖ Martin Fowler, Kendall Scott, 「UML Distilled: A Brief Guide to the Standard Object Modeling Language, 2nd Edition」, Addison-Wesley Pub Co (邦訳: 「UMLモデリングのエッセンス-標準オブジェクトモデリング言語入門 第2版」, 翔泳社)
- ❖ 日経ソフトウェア(編): 「ゼロから学ぶソフトウェア設計」, 日経BP

47