

# ソフトウェア工学

## 12: UML補遺 — クラス図の拡張的概念

理工学部 経営システム工学科  
庄司 裕子



## 今回のテーマ

### UML補遺 —

#### クラス図の拡張的概念の一部

☞ 必須ではないが、ある状況では有効な補足  
的概念

- ステレオタイプ
- 単一/多重分類と静的/動的分類
- 限定子付き関連
- 関連クラス

2

## 1. ステレオタイプ

## ステレオタイプ

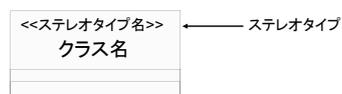
- UMLの基本的な拡張メカニズム
  - UMLには定義されていないが類似したモデリング要素が必要な場合、その要素をUMLの定義済み要素のステレオタイプとして扱う
- 例: インタフェース
  - インタフェースはメソッド本体と属性を持たず、パブリックな操作(仕様)だけを持つ特殊なクラス
  - クラスのステレオタイプとして定義されている

4

## ステレオタイプ(続)

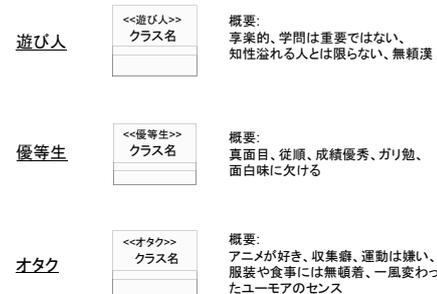
- 問題領域(ドメイン)の言葉を用いて表現される基本的なビルディングブロック(構築部品)のようなものを新たに導入したほうがよい場合がある

⇒それがステレオタイプ



5

## 日常的な例: 人間のステレオタイプ



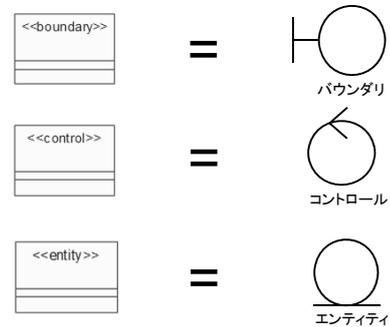
6

### 有用なステレオタイプ：分析クラス

- 分析クラスは、システムのオブジェクトモデルの「最初の切り口」を獲得するために使われる
- 分析クラスは、主として機能的な要求を扱い、問題領域のオブジェクトをモデリングする
- システムの3つの側面を分離し、一連の責務をカプセル化した3種類の分析クラスを識別する：
  - システムとアクター間の境界： **バウンダリ** クラス
  - システムの制御ロジック： **コントロール** クラス
  - システムで扱われる情報： **エンティティ** クラス

7

### 分析クラスの表記法



8

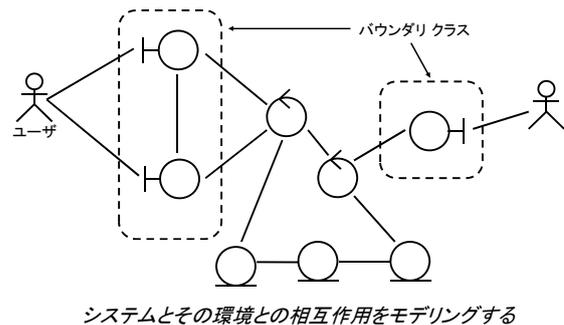
### バウンダリ クラスとは

- システムと外部アクターの間で介在するオブジェクトをモデリングするもの
  - ユーザ インタフェース
  - デバイス インタフェース
  - 外部システム インタフェース
- 環境に依存する
  - GUI に依存
  - 通信プロトコルに依存



9

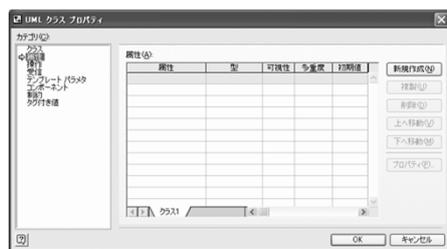
### バウンダリ クラスの役割



10

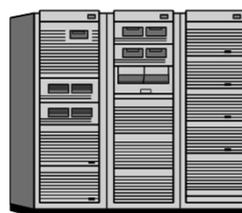
### バウンダリ クラスの例： ユーザ インタフェース

クラス プロパティ定義画面



11

### バウンダリ クラスの例： 外部システム インタフェース

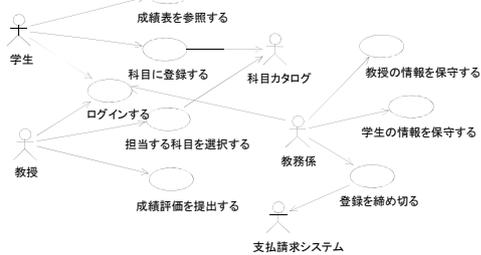


顧客データベースシステム

12

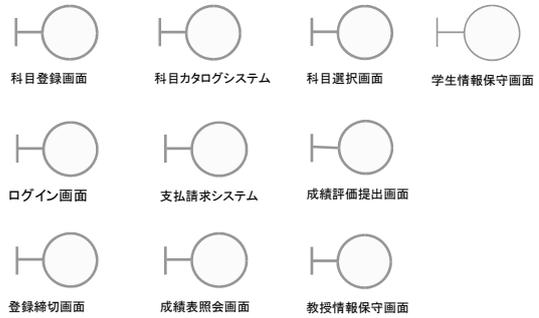
## バウンダリ クラスの発見方法

- ユースケースとアクターの組ごとにバウンダリ オブジェクトが 少なくとも1つ 存在しなければならない



13

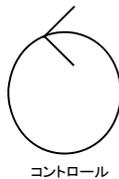
## 発見されたバウンダリ クラス



14

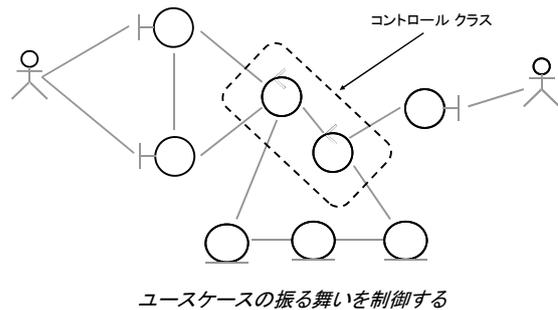
## コントロール クラスとは

- ユースケースの振る舞いを制御するもの
- ユースケースの働きを他のクラスに委譲する
- ユースケースに依存し、環境には依存しない(サポート対象のユースケースがなくなった後は存続しない)



15

## コントロール クラスの役割



16

## コントロール クラスの利用上の注意

- ☞ コントロール クラスは実行順序の決定だけを行う
- ☞ コントロール クラスは何かを行うように他のクラスに命令する。命令以外は何も行ってはいけない
- ☞ コントロール クラスを用いて、バウンダリ クラスとエンティティ クラスを切り離す

17

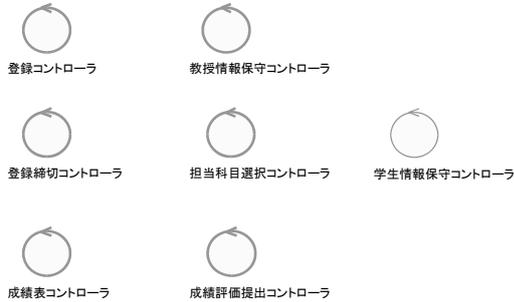
## コントロール クラスの発見方法

- 複雑なユースケースごとにコントロール クラスが 1つ ずつ存在する



18

## 発見されたコントロール クラス



19

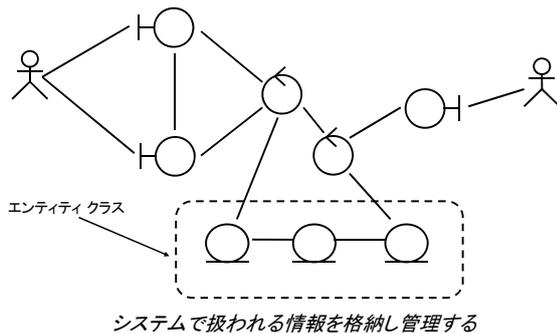
## エンティティ クラスとは

- システムの主要概念のうち、相互作用を開始しないパッシブ（受動的）なオブジェクトをモデリングするもの
- 通常、長期にわたって存続する永続的な情報をモデリングする
- 環境には依存しない
- 複数のユースケースに参加できる



20

## エンティティ クラスの役割



21

## エンティティ クラスの発見方法

- ユースケースに記述されている名詞および名詞句を調べることで、エンティティオブジェクトを洗い出す
- 見つかった名詞は以下のいずれかの可能性がある
  - オブジェクト
  - オブジェクトの状態を記述したもの
  - 外部の実体やアクター
  - 上記のいずれにも該当しないもの
- オブジェクト以外のものや重要でないオブジェクトを多数洗い出すおそれがあるので、名詞のリストを選別する必要がある

22

## 名詞を選別する際の注意事項

- 複数の用語が同じオブジェクトを指していることがある
- 1つの用語が複数のオブジェクトを指していることがある
- いかなる名詞も「動詞化」でき、いかなる動詞も「名詞化」できる

23

## サンプルのユースケース シナリオ

- 「スケジュールを作成する」のシナリオ

太郎が学生番号 123-523449 を入力すると、システムはその番号が正しいかどうかチェックする。システムはどの学期かを尋ねる。太郎は今学期を指定し、新しい履修スケジュールの作成を選択する。

履修可能な科目のリストから、太郎は必修科目である「英語学(Ⅰ)」、「数理経済学基礎(Ⅰ)」、「ソフトウェア工学」、および「情報数学」を選択する。そのあと、彼は選択科目である「異文化コミュニケーション論」と「Java プログラミング入門」を選択する。

システムは、学生履修記録を調べることで、太郎が必要な履修条件をすべて満たしていることを確認し、太郎を科目履修者名簿に登録する。システムは処理が完了したことを知らせる。システムは学生履修スケジュールを出力し、6科目分の授業料請求情報を支払請求システムに送信し処理させる。

- ◆ この記述から名詞を洗い出してみよう

24

## 「スケジュールを作成する」の名詞リスト

|               |                 |
|---------------|-----------------|
| 太郎            | 学生番号 123-523449 |
| システム          | 番号              |
| 学期            | 今学期             |
| 新しい履修スケジュール   | 履修可能な科目のリスト     |
| 必修科目          | 英語学(Ⅰ)          |
| 数理経済学基礎(Ⅰ)    | ソフトウェア工学        |
| 情報数学          | 選択科目            |
| 異文化コミュニケーション論 | Java プログラミング入門  |
| 必要な履修条件       | 学生履修記録          |
| 科目履修者名簿       | 処理              |
| 学生履修スケジュール    | 授業料請求情報         |
| 6 科目          | 支払請求システム        |

◆ 除外すべき名詞/名詞句は何か

25

## 名詞ごとの選別結果

| 名詞              | 選別結果                    |
|-----------------|-------------------------|
| 太郎              | 除外(アクター)                |
| 学生番号 123-523449 | 除外(学生のプロパティ)            |
| システム            | 除外(構築対象)                |
| 番号              | 除外(学生番号 123-523449 と同じ) |
| 学期              | 除外(状態 - 選択時)            |
| 今学期             | 除外(学期と同じ)               |
| 新しい履修スケジュール     | オブジェクト候補                |
| 履修可能な科目のリスト     | オブジェクト候補                |
| 必修科目            | 除外(選択された科目の状態)          |
| 英語学(Ⅰ)          | オブジェクト候補                |
| 数理経済学基礎(Ⅰ)      | オブジェクト候補                |
| ソフトウェア工学        | オブジェクト候補                |
| 情報数学            | オブジェクト候補                |

26

## 名詞ごとの選別結果(続き)

| 名詞             | 選別結果               |
|----------------|--------------------|
| 選択科目           | 除外(選択された科目の状態)     |
| 異文化コミュニケーション論  | オブジェクト候補           |
| Java プログラミング入門 | オブジェクト候補           |
| 必要な履修条件        | 除外(他の識別済み科目と同様の科目) |
| 学生履修記録         | オブジェクト候補           |
| 科目履修者名簿        | オブジェクト候補           |
| 処理             | 除外(日本語の言い回し)       |
| 学生履修スケジュール     | 除外(スケジュールと同じ)      |
| 授業料請求情報        | オブジェクト候補           |
| 6 科目           | 除外(支払請求システムに必要な情報) |
| 支払請求システム       | 除外(アクター)           |

27

## 「スケジュールを作成する」のエンティティ オブジェクト候補(選別済)

| オブジェクト         | 概要                    |
|----------------|-----------------------|
| 新しい履修スケジュール    | ある学期に学生が履修する科目のリスト    |
| 履修可能な科目のリスト    | ある学期に開講される全科目のリスト     |
| 英語学(Ⅰ)         | ある学期に開講される科目          |
| 数理経済学基礎(Ⅰ)     | ある学期に開講される科目          |
| ソフトウェア工学       | ある学期に開講される科目          |
| 情報数学           | ある学期に開講される科目          |
| 異文化コミュニケーション論  | ある学期に開講される科目          |
| Java プログラミング入門 | ある学期に開講される科目          |
| 学生履修記録         | 学生がこれまでの学期に履修した科目のリスト |
| 科目履修者名簿        | 特定の科目を履修する学生のリスト      |
| 授業料請求情報        | 「支払請求システム」アクターに必要な情報  |

28

## エンティティ クラスの特定

- エンティティ オブジェクトをクラスに分類する
  - ▷ 分類は、構造や振る舞いの類似性に基づいて行う
- 検討されるシナリオが増えるにつれて、識別されるクラスが変わる可能性がある
- すべてのユースケース シナリオの評価が完了した後は、エンティティ クラスが識別されないなければならない

29

## 発見されたエンティティ クラス

- 授業カタログ: ある学期に開講される全科目のリスト  
- 履修スケジュール: ある学期に学生が履修する科目のリスト  
- 学生履修記録: 学生がこれまでの学期に履修した科目のリスト  
- 科目: ある学期に開講される科目  
- 授業料請求情報: 「支払請求システム」アクターに必要な情報  
- 科目履修者名簿: 特定の科目を履修する学生のリスト  

30

### 分析クラスのチェックポイント

- 各バウンダリクラスはユースケースとアクターの組を表しているか
- 各ユースケースにはコントロールクラスが割り当てられているか
- 開発対象システムの主要概念ごとに、対応するエンティティクラスが発見されているか

31

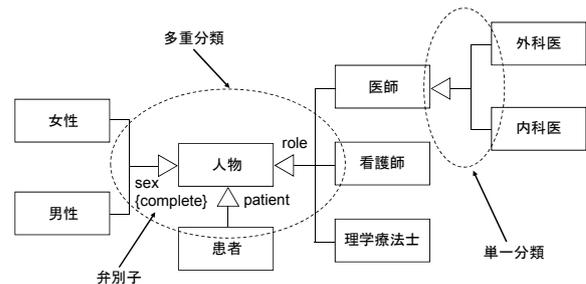
## 2. 単一/多重分類と静的/動的分類

### 単一分類 v.s. 多重分類

- 単一分類 (single classification)
  - オブジェクトはスーパータイプから継承する単一のタイプ (直接クラス) に属する
- 多重分類 (multiple classification)
  - 1つのオブジェクトが複数のタイプ (直接クラス) の直接インスタンスになることができる
  - その特化の次元を弁別子 (discriminator) と呼び、汎化シンボルにそのラベルを付ける
  - 複数のサブタイプで同じ弁別子を共有できるが、同じ弁別子を持つサブタイプはすべて排他的。1つの弁別子に属するサブタイプが完全に列挙されている場合、その弁別子に {complete} という制約を付ける

33

### 単一分類と多重分類の例



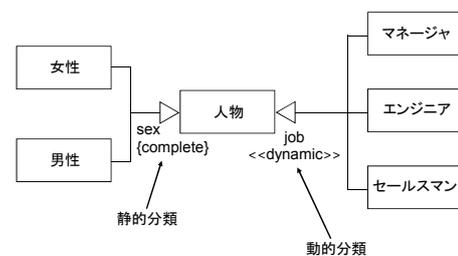
34

### 動的分類 v.s. 静的分類

- 動的分類 (dynamic classification)
  - オブジェクトは属するタイプ (直接クラス) を実行時に変更できる
  - 例: 銀行口座が借り越し状態になると、実質的に振る舞いが変わる (すなわち、タイプが変わる)
- 静的分類 (static classification)
  - 上記の変更はできない

35

### 静的分類と動的分類の例



36

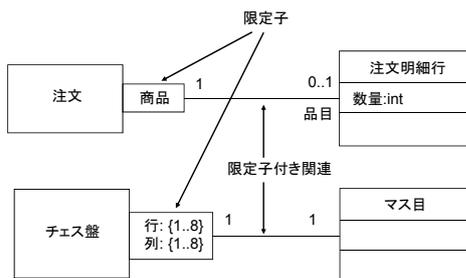
### 3. 限定子付き関連

### 限定子(qualifier)とは

- (2項) 関連についての属性または属性リスト用のスロットのこと
- その関連によって1つのオブジェクト(「限定子付きオブジェクト」と呼ぶ)に関係づけられているオブジェクト群から、一意なオブジェクトまたはオブジェクト群を選択するのに使われる
- 限定子の値に基づいて選択されるオブジェクトを「ターゲットオブジェクト」と呼び、限定子は常に、ターゲット方向に「多」の多重度を持つ関連に作用
- 限定子で修飾された関連を限定子付き関連(qualified association)と呼ぶ

38

### 限定子付き関連

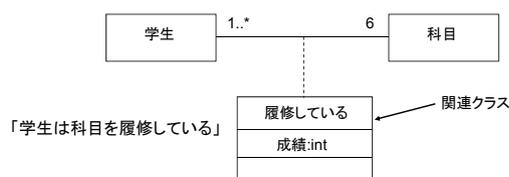


39

### 4. 関連クラス

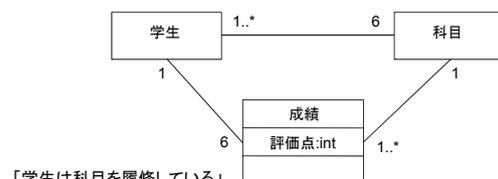
### 関連クラス

- 関連クラス (association class) を使うと、
- 関連に属性、操作などの特性を追加できる
  - 関連に参加する任意の2オブジェクト間に存在する関連クラス インスタンスは1つに絞られる



41

### 関連クラスを使わない方法



☞ 同じ学生が同じ科目を複数回履修することが可能になってしまう

42

## まとめ

- 必須ではないものの、ある状況では有効な、クラス図の拡張的概念の一部をUML補遺として解説した
  - ステレオタイプ
  - 単一/多重分類と静的/動的分類
  - 限定子付き関連
  - 関連クラス
- 特に、分析作業に有効なステレオタイプである3種類の分析クラスを詳説した

43

## 参考文献

- 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)
- 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)

44