

ソフトウェア工学

09: UMLモデリング(Ⅳ) — システムの動的側面(オブジェクトの状態変化)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(Ⅳ) — システムの動的側面(オブジェクトの状態変化)のモデリング

- > オブジェクトの状態変化
- > ステートチャート図
- > アクティビティ図

2

UMLダイアグラム

- | | |
|--|--|
| <ul style="list-style-type: none"> ■ システムの静的(構造的)側面 ✓ 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"> ■ システムの動的側面 ✓ ユースケース図 ✓ シーケンス図 ✓ 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 |
|--|--|

3

システムの動的側面(オブジェクトの状態変化)のモデリング

- ⇨ オブジェクトの状態変化
- ステートチャート図
- アクティビティ図

4

オブジェクト指向的システム観

- システム
 - ⇨ 特定の責務と状態を備えたオブジェクトの集合
- システムの処理
 - ⇨ システムを構成するオブジェクト同士のメッセージ通信
- 処理の結果
 - ⇨ システムを構成する個々のオブジェクトの状態変化

5

オブジェクトの状態変化のモデリング

- UMLでは、システムの処理はシステムを構成するオブジェクト間の相互作用(メッセージ通信)としてモデリングされ、相互作用図で表現される
- 相互作用の結果、個々のオブジェクトには状態変化が起きる
- ⇨ 1つのオブジェクトに着目して、その状態変化を表現するのがステートチャート図

6

オブジェクトの状態とその実装

- オブジェクトの状態は、オブジェクトがその**存続期間中に取り得る状況の1つで、通常、時間の経過と共に変化する**
- 通常、一群の属性、それらの属性値、および他のオブジェクトとの間に存在し得るリンクで実装される
 - > 「状態」属性(または属性群)で定義されるわけではなく、オブジェクトの属性およびリンク全体で定義される

7

復習：属性とは

- 属性はクラスのプロパティであり、名前を持ち、プロパティのインスタンスが保持する値の範囲を記述したもの(データ定義)である
- クラスには任意の数(0個以上)の属性を定義できる



8

復習：属性の表現

- 名前と型を指定する(必要であればデフォルト値も)
 - > <属性名> : <データ型> = <デフォルト値>
- 実装言語およびプロジェクトの命名規則に従う
- 型には実装言語の要素データ型を使用する
 - > 組み込みデータ型、ユーザ定義のデータ型、またはユーザ定義のクラス
- 可視性を指定する
 - > public: "+"
 - > private: "-"
 - > protected: "#"

9

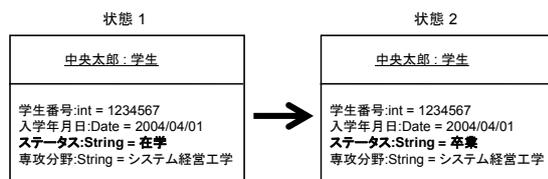
クラスの属性とオブジェクトの状態

- クラスの操作の実行に必要な属性は、状態の識別と同時に識別される
- 属性はクラス インスタンス(オブジェクト)に関する情報の格納先であり、クラス インスタンスの**状態**を表現するために使用されることが多い
 - ☞ それ以外に、実装用の属性もある

10

オブジェクトの状態変化

- 中央太郎さんの「ステータス」属性が「在学」から「卒業」に変わったら、「中央太郎」オブジェクトの状態は変化する



11

システムの動的側面(オブジェクトの状態変化)のモデリング

- ✓ オブジェクトの状態変化
 - ⇒ ステートチャート図
 - アクティビティ図

12

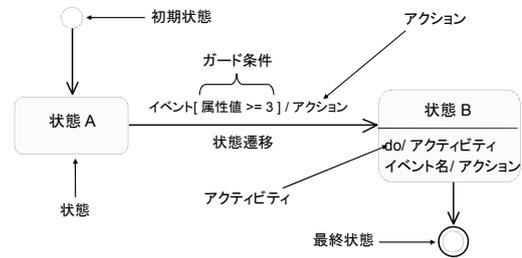
ステートチャート図とは

- ステートチャート図(状態図)は状態マシンを示すもので、以下を記述する
 - オブジェクトが取り得る状態のシーケンス
 - オブジェクトを特定の状態に遷移させるイベントと条件
 - 特定の状態に遷移したときに発火するアクション



13

ステートチャート図の主な構成要素



14

状態の描き方

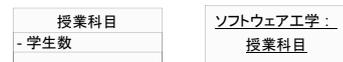
- 状態は角の丸い四角形で表現される
- 以下のものを四角形の中に書く
 - 状態名
 - イベント名/アクション (入状/退状アクションの場合、状態がイベントに対して遷移を伴わないアクションで応答する場合)
 - do/アクティビティ (状態にアクティビティが関連付けられている場合)



15

状態と属性

- 状態は特定の属性の値によって識別される場合がある



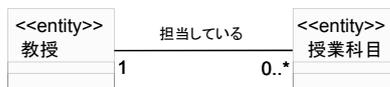
- 授業科目あたりの学生数の上限は 10 名



16

状態とリンク

- 状態は、特定のリンクの存在によって識別される場合もある
- (例) 教授クラスのインスタンスが取り得る状態は 2 つ
- 授業科目へのリンクが存在するときは「勤務中」
 - 同リンクが存在しないときは「研究休暇中」



17

初期状態と最終状態

- 初期状態は、作成された時点のオブジェクトが取る状態である
 - 初期状態は必須
 - 初期状態は 1 つだけ定義できる
 - 初期状態は塗りつぶされた円で表される
- 最終状態は、オブジェクトの寿命の終わりを示す
 - 最終状態は省略可能
 - 最終状態は複数存在してもよい
 - 最終状態は、内側が塗りつぶされた二重の円で表される



18

イベントとは

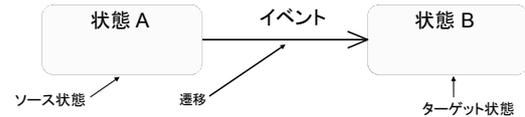
- 時空間で発生する注目に値する出来事の仕様のこと
 - 状態遷移をトリガする刺激の発生を表す (例)
 - 授業科目への学生の追加
 - 新しい授業科目の作成



19

遷移とは

- 元の状態 (ソース状態) から新しい状態 (ターゲット状態) への変化のことで、何らかの刺激の結果として発生する
 - ターゲット状態がソース状態と同じ場合がある (自己遷移)
- 遷移はイベントに反応して発生する場合がある
- イベントによってラベル付けすることができる



20

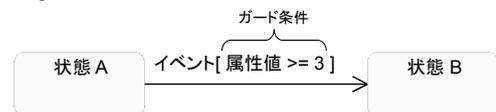
遷移ラベルの構文

- 3つの部分から成る
 - イベント [ガード条件] / アクション
 - イベント ::= イベント名 (引数*)
- 各部分は省略可能 (必ずしも明記する必要はない)

21

ガード条件とは

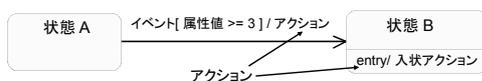
- トリガ イベントの受信によって遷移がトリガされたときに評価される論理式
 - 式が「真」であると評価された場合、遷移が発火する
 - 式が「偽」であると評価された場合、遷移は発火しない



22

アクションとは

- 実行可能な最小単位の計算処理で、結果としてモデルの状態変化を引き起こすか、値を返す
 - 遷移と関連付けられる
 - 完了にそれほどの時間を要さない
 - 割り込み不可



23

アクションの種類

- 状態マシンの所有者やその他の可視オブジェクトに対しての操作の呼び出し
- 別のオブジェクトの作成または破棄
- 別のオブジェクトへのシグナル送信
 - シグナル送信の場合は、シグナル名の先頭にキーワード「send」を付けて示す
- 入状 / 退状時アクション

24

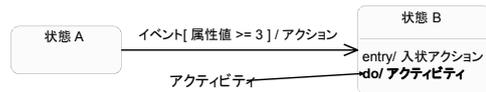
入状／退状時アクション

- 入状時アクション
 - 状態に入るたびに実行されるアクション
- 退状時アクション
 - 状態から抜けるたびに実行されるアクション
- 入ってくる遷移や出て行く遷移に個別にアクションを関連付けなくても、この動作を容易に実現できる
- 引数やガード条件は指定できない
 - モデル要素の状態マシンの最上位にある入状時アクションには、要素が作成されたときに状態マシンが受け取る引数を表すパラメータを指定できる

25

アクティビティとは

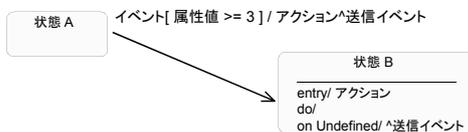
- 状態マシン内部で継続的に実行される非アトミックな処理のこと
 - 状態と関連付けられる
 - 完了に時間を要する操作
 - 状態に入った時に開始される
 - 完了まで実行されることもあれば、外に出ていく遷移によって割り込まれることもある



26

送信イベント

- あるイベントが、別のイベントの送信をトリガする場合がある
- アクティビティは別のオブジェクトにイベントを送信することもできる



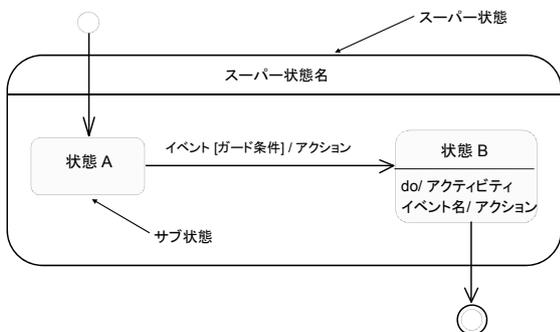
27

状態の階層化

- ステートチャート図は非常に複雑になることがある
- サブ状態(入れ子状態)を使って、複雑なダイアグラムを簡潔にすることができる
- スーパー状態は、サブ状態と呼ばれる入れ子状態を内包する状態である
- 入れ子の深さに制限はない
- 状態を入れ子にすることにより、視覚上の複雑さが大幅に緩和され、より大規模で複雑な問題のモデリングが可能になる

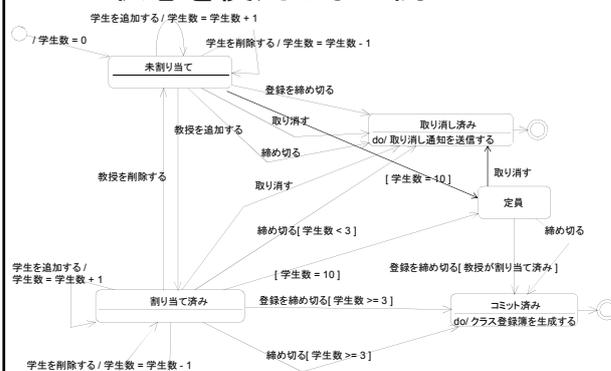
28

スーパー状態とサブ状態



29

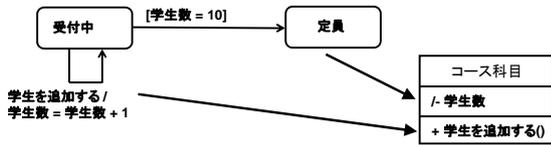
サブ状態を使用しない例



30

状態チャートとその他のモデル要素の対応

- イベントは操作に対応する場合がある
- 状態は属性を使って表現されることが多い



37

システムの動的側面(オブジェクトの状態変化)のモデリング

- ✓ オブジェクトの状態変化
- ✓ ステートチャート図
- ⇨ アクティビティ図

38

アクティビティ図とは

- アクティビティ(何かを実行している状態)の流れ/変化を表す
- 基本的にはフローチャートであり、アクティビティ間の制御フローを示す
- 以下に有効
 - > ユースケース内のアクティビティの表現
 - > ワークフローの理解
 - > 複雑な逐次アルゴリズムの記述
 - > マルチスレッドアプリケーションの記述

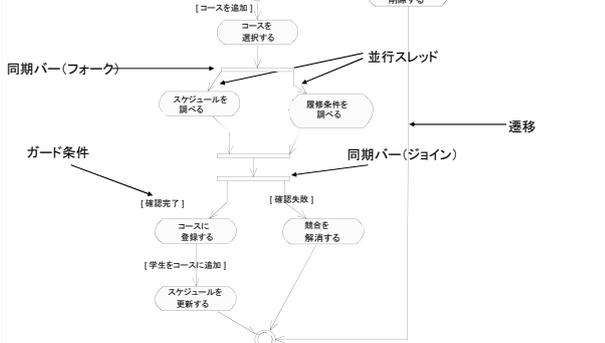
39

アクティビティ状態とは

- 以下の特徴を備えた継続的に進行する非アトミックな実行のこと
 - > 分解可能
 - > 割り込み可能で、完了するのにある程度の時間がかかる
- アクティビティ状態は、ワークフロー内でのアクティビティまたは作業ステップの実行を表す
- ステートチャート図におけるアクティビティと混同しないように

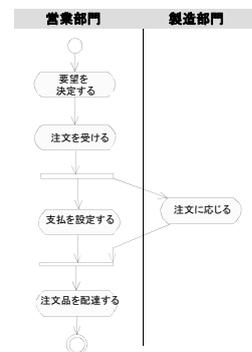
40

(例) 分岐



41

レーン



42

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

43