

UMLの使い方

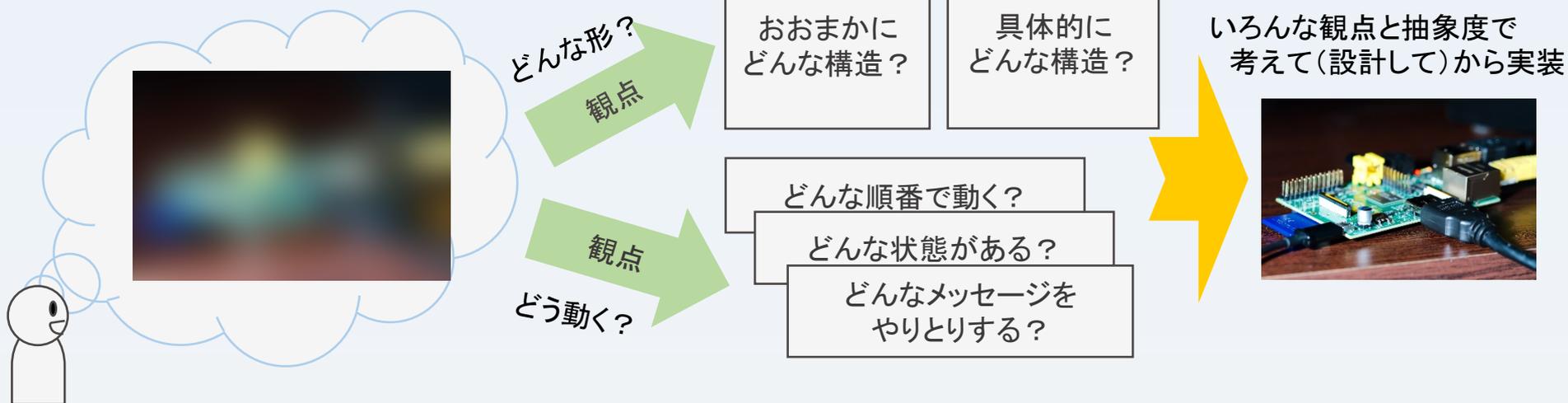
Unified Modeling Language

細合 晋太郎

高瀬 英希

そもそもモデリングとは？

- システムは超複雑
いきなり作る(考える)のは難しい



- ある観点だけのある抽象度で抽出したもの → モデル
- 世にあるほとんどの～図はモデルと言ってもいい
 - Ex) 上から見た構造 → 上面図
 - Ex) 粘土で形だけ → クレイモデル
 - Ex) 音階とタイミングで音楽を分解 → 楽譜

また、設計者・実装者間で意思統一を図るためにもモデリングは重要

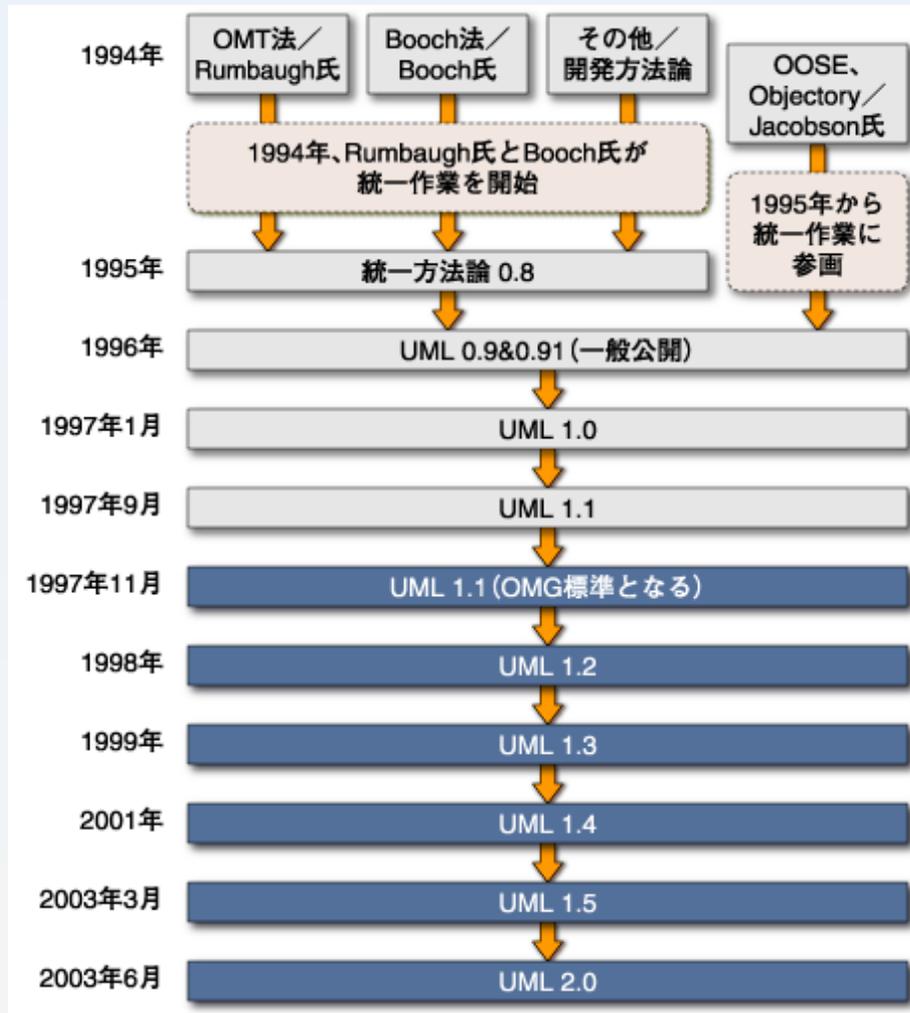
構造モデルと振舞いモデル

- 観点は大きく分けると構造と振舞いに大別できる。
- どんな形をしているのか？ どんな動きをするのか？
- UMLでは、構造と振舞いのモデルが様々な抽象度と観点で定義されている。

UMLモデルは厳密に定義されている。

- UMLの仕様書 (Super Structure)
<http://www.omg.org/spec/UML/2.5/PDF/>

モデリング言語の統一



2015/07/31 現在

<http://www.omg.org/spec/UML/>

<http://thinkit.co.jp/free/compare/12/1/>より転載

UML Diagrams (version 2.5)

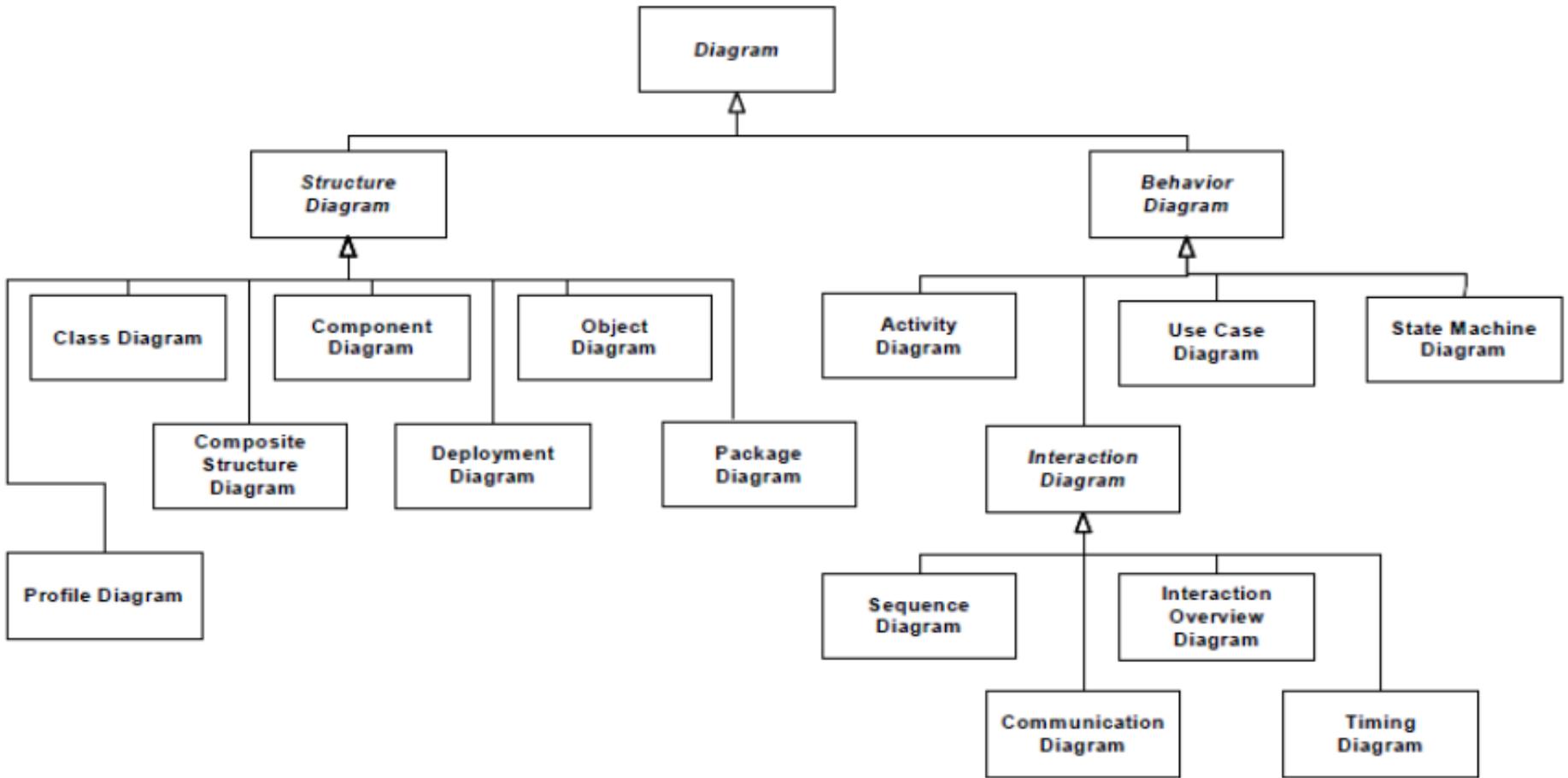


Figure A.5 The taxonomy of structure and behavior diagrams

<http://www.omg.org/spec/UML/2.5/PDF/> pp683

構造モデル

- Deployment Diagram : 配置図
- Component Diagram : コンポーネント図
- Composite Structure Diagram : 複合構造図
- Package Diagram : パッケージ図
- **Class Diagram : クラス図**
- Object Diagram : オブジェクト図



- Profile Diagram : プロファイル図 これは別観点

振舞いモデル

観点：

- Use Case Diagram : ユースケース図
 - Activity Diagram : アクティビティ図
 - **State Machine Diagram : ステートマシン図**
 - Interaction Diagram : 相互作用図
 - Sequence Diagram : シーケンス図
 - Communication Diagram : コミュニケーション図
 - Interaction Overview Diagram : 相互作用概念図
 - Timing Diagram : タイミング図
- 外界との関係, 要求
システムフロー
状態とその遷移
- オブジェクト間フロー
オブジェクト相互関係
図間のオーバービュー
動作タイミング

振舞いモデルは見たい観点がそれぞれ異なる

すべてのUML図を使う必要はない

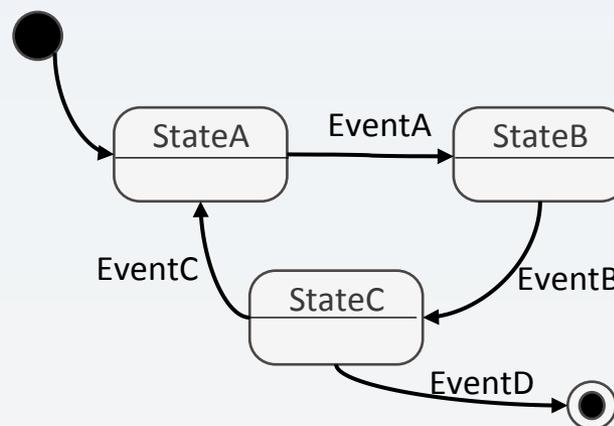
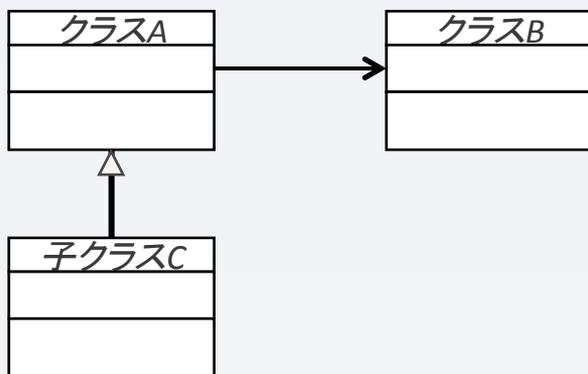
- UMLの各図は、観点と抽象度のフレームを与えてくれるもの。
 - 必要な図を取捨して選択すればよい
- 図の用途も様々ある
 - 設計分析
 - ステークホルダー間の意思統一
 - エビデンス
 - . . .
- では、どのようなモデルを選択すればよいか？
 - UMLの開発プロセスを利用してみよう

UMLのプロセス

- 多くのプロセスでは、トップダウンにモデルを記述していく。
- ICONIXプロセスでは、以下の順でモデルを詳細化していく
 - ユースケース図・ユースケース記述
 - ロバストネス図
 - ドメインモデル
 - クラス図
 - シーケンス図
- ただし、何にでも適用できる唯一のプロセスはない
- まずは、既存のプロセスに当てはめてみて、徐々に対象に合わせたモデルを選択し、プロセスを改善していく。

LED-Campにおける設計プロセス

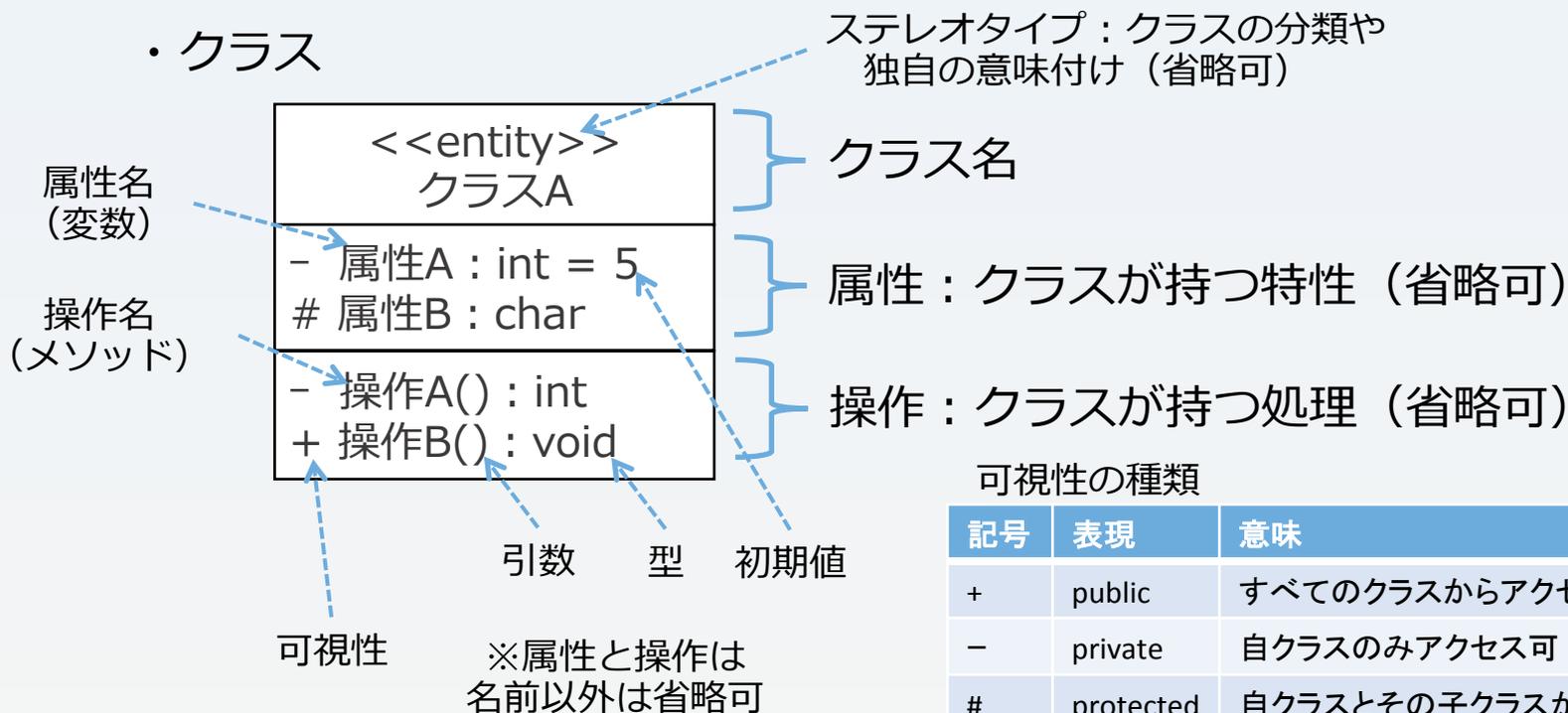
- LED-Camp3の実習では、
構造モデルとしてクラス図
振る舞いモデルとしてステートマシン図
を使用します。
 - 今回の開発教材の規模と機能実現に向いている
 - (小規模であれば) この2図だけでシステム全体の構成を俯瞰するには十分



クラス図

- システムの静的な構造を表現するためのモデル
- オブジェクトの情報を抽象的に定義したもの
 - クラスにはデータや処理に対応する属性や操作の情報を表現できる
 - 個々のクラス間のつながりとして、様々な種類の関係を表現できる

• クラス

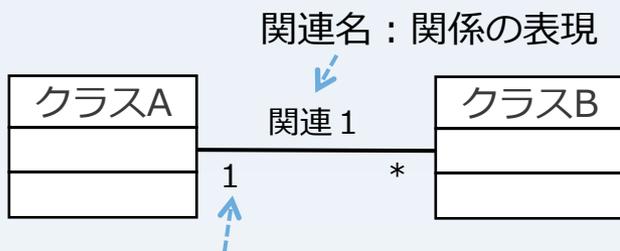


可視性の種類

記号	表現	意味
+	public	すべてのクラスからアクセス可
-	private	自クラスのみアクセス可
#	protected	自クラスとその子クラスからアクセス可
~	package	同一パッケージ内のクラスからアクセス可

クラス図

- ・ 関連：クラス同士の関係



多重度：関係の個数

多重度	意味
1	1のみ
*	0以上
0..1	0 or 1
1..*	1以上
3..6	3~6

- ・ 関連の種類

→ 単方向関連

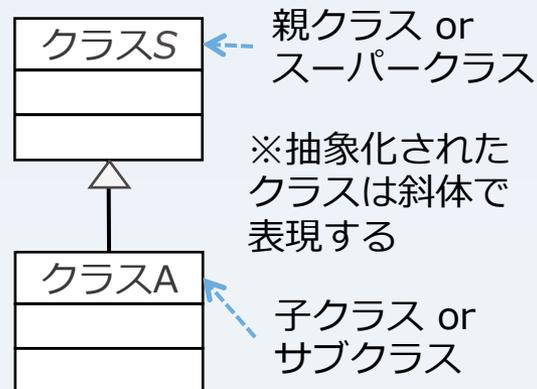
↔ 双方向関連

- ・ 誘導可能性：関連の一方方向性
一方からもう一方を参照可能か

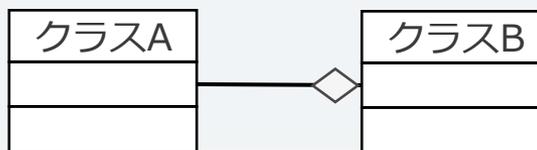


誘導不可 誘導可能

- ・ 汎化：クラスの抽象化



- ・ 集約：全体-部分の関係
より強い関係を表すコンポジションもある

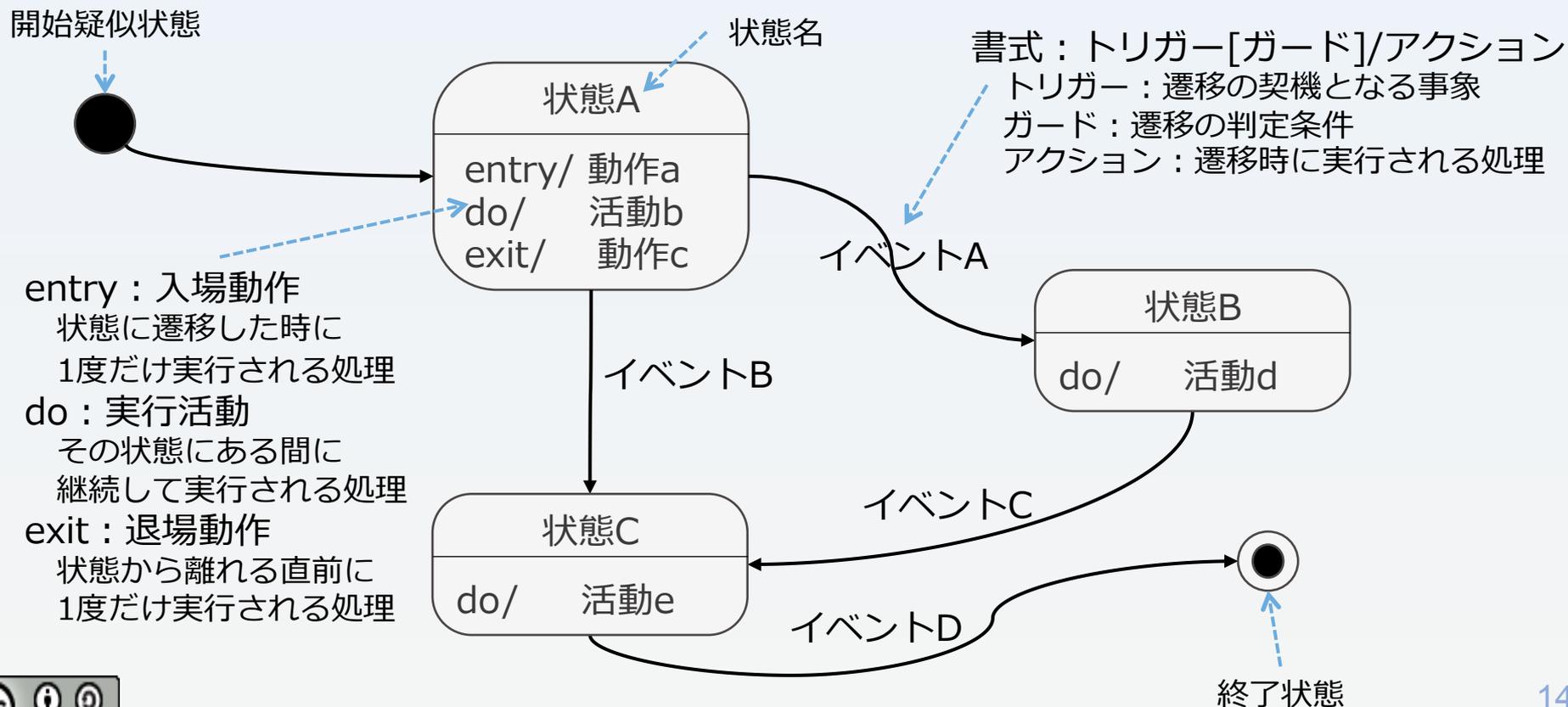


クラス図のモデリング

- 一つのクラスに一つの責務
 - 複数の責務を持つクラスは分割する.
 - 1つのクラスが複数の責務を持つと, 処理が複雑になる, 影響範囲がわかりづらくなるなど, 多くの場合で問題が多くなる.
- 継承関係・依存関係はできるだけシンプルに
 - 多重継承を持たない
 - 循環参照を持たない
 - 所有関係を考慮する (集約)
- 今回のみの制約: MDDプラグインが未対応
 - 集約とコンポジションは反映されない
 - 関連と解釈されたソースコードを出力する
 - 双方向の関連は持たない
 - 循環参照になりやすいため

ステートマシン図

- クラス（オブジェクト）の状態と遷移を表現するためのモデル
 - 状態遷移は、ある状態から別の状態への遷移で表現できる
 - 遷移を起こす契機や遷移を行うかの判断条件を表記できる



ステートマシン図のモデリング

- 基本的に、イベント、ガード条件、entryアクションで記述していく。
 - 遷移アクション、doアクション、exitアクションはとりあえず使わない。
 - トリガーが発生してもガード条件が満たされない場合は遷移しない
 - トリガー省略時は状態のexitアクションが実行されたら自動的に遷移する
- 非決定な遷移を含まない
 - 同じイベント・ガードで複数の遷移を持たない
 - 同じイベントが来た際に、次の状態が非決定とならないように
- フローチャートにしない
 - ステートマシンは状態を捉える図なので、ただ処理手順を書き並べるような図にならないように

モデリング全体の注意点

- 記法・ルールを守る
 - UMLに厳密に沿う必要はないが、読む可能性のある人の中でルールの統一ができていること。
- 名前付けは的確に
 - ステークホルダーが容易に理解できるものを心がける
- モデル図間，モデル - コード間の整合性
 - 上流から徐々にブレークダウンしていても，その間の整合性が欠けると破綻してしまう。修正した場合などは，整合性を確認すること
- モデル図のリファクタリング
 - モデル要素やテキストの位置，線の引き方でもモデルの見やすさは変わってくる。綺麗なモデルとなるように，心がける

参考になるWebページ

UMLの詳細については、下記リンクの記事をご参照ください。

- 日経BP UMLモデリングの基礎
 - <http://itpro.nikkeibp.co.jp/article/lecture/20061204/255756/>
 - Part1, Part2, Part7のUML・モデリングについてが分かりやすいと思います。
- マイナビニュース ゼロから始めるUMLモデリング講座
 - http://news.mynavi.jp/series/uml_zero/menu.html
 - 1, 2, 3, 6, 7, 15, 16, 22回
 - こちらもUMLの基礎です。一回あたりの量は多くないので、さらりと読めます。
- 豆蔵ソフト工学ラボ
 - http://labo.mamezou.com/special/sp_006/
 - 組込みモデリングの勘所についてです。
- OMG UML Specification(Superstructure)
 - <http://www.omg.org/spec/UML/2.5/>
 - UMLの本家の仕様です。読む必要はありませんが、存在は知っておいてください

理解度チェック

この資料では、以下について解説しました。

- UMLによるモデリングの考え方
- クラス図：システムの静的な構造を表現するためのモデル
 - クラスの属性および操作を表現する記法
 - クラス間の関係を表現する記法
- ステートマシン図：状態と遷移を表現するためのモデル
 - 状態の処理および遷移の条件を表現する記法
- クラス図とステートマシン図によるモデリングの勘所と注意点

当日実習までに全て理解できているようにしてください。

質問や不明な点があれば、実行委員会までご連絡ください。