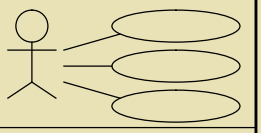
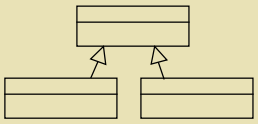
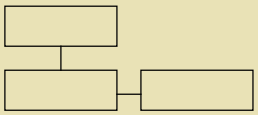
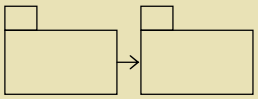
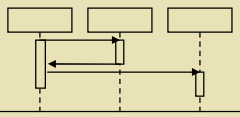
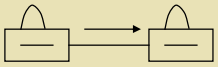
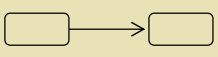
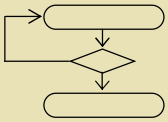
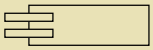
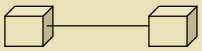
以降のページはHPで公開しているため、書き写し不要

UMLの各図

ダイアグラム	役割	開発フェーズ	図
ユースケース図	システムの要件定義 アクターとシステム、また外部システムとの関係を明記	分析(要件定義)	
クラス図	システムの静的な部分の設計図	分析・設計	
オブジェクト図	クラス図から作られるオブジェクト(インスタンス)の具体的な構成図	分析・設計	
パッケージ図	パッケージの階層関係と依存関係を明記 (パッケージ： 共通部品の分類と配置)	分析・設計	

UMLの各図

ダイアグラム		役割	開発フェーズ	図
相互作用図	シーケンス図	オブジェクト間のメッセージの受渡しを時系列で表記	分析・設計	
	コラボレーション図	オブジェクト間の関係やメッセージ受渡しを構造で表記	分析・設計	
ステートチャート図		オブジェクトの状態や状態遷移の条件を表記	分析・設計	
アクティビティ図		フローチャート	分析・設計	
コンポーネント図		ソースファイルや共通部品の配置と依存関係を明記	設計	
配置図		ハードウェア構成と、ソフトウェア・コンポーネントの配置図	設計	

UML表記の拡張

UMLは次のように表記を拡張して、利用しやすくすることができる

- ステレオタイプ
  - クラス図などで、モデル要素の意味を拡張するもの。ギルメット << >> によるラベル表記と、アイコン表記がある。

ラベル表記とアイコン表記　　ラベル表記とアイコン表記

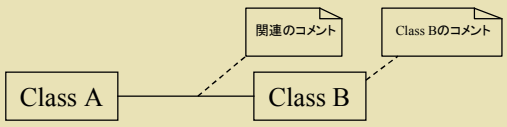
- ステレオタイプ一覧

ステレオタイプ名	拡張元要素	意味
<< actor >>	クラス	ユースケースで相互作用するユーザや外部システム
<< include >>	依存	あるユースケースが、必ず別のユースケースを実行するなど依存している場合
<< extend >>	依存	ある特定の条件（拡張点）で、発生する振る舞い
<< interface >>	クラス	クラス、コンポーネントのサービスを仕様化するための操作の集合
<< create >>	メッセージ	ターゲットオブジェクトがメッセージによって生成される
<< destroy >>	メッセージ	ターゲットオブジェクトがメッセージによって消滅される
<< bind >>	依存	パラメータに値を設定してインスタンス化する
<< framework >>	パッケージ	パッケージが主にパターンから生成されることを指定する
<< executable >>	コンポーネント	ノード上で実行可能なコンポーネントであることを指定する
<< library >>	コンポーネント	ライブラリであることを示す（静的 or 動的）。
<< file >>	コンポーネント	ソースコード　又はデータが入っているファイルを示す。
<< document >>	コンポーネント	文書を表すコンポーネントを示す。

UML表記の拡張

UMLは次のように表記を拡張して、利用しやすくすることができる

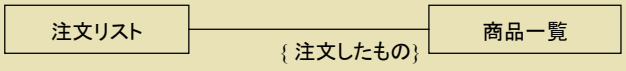
- ノート
  - クラスや関連などのモデルに対して説明を加える



- タグ付き値
  - モデル要素の持つプロパティーを { キー = 値, キー = 値, ･･･}で表す

右は簡略表記

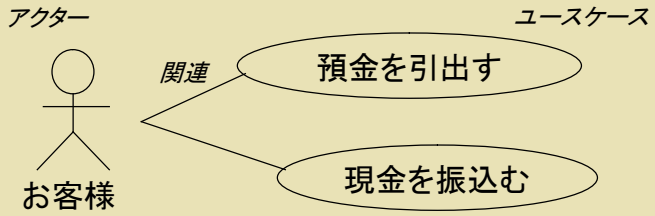
- 制約
  - 要素に付加されている制約を { } で表現する



ユースケース

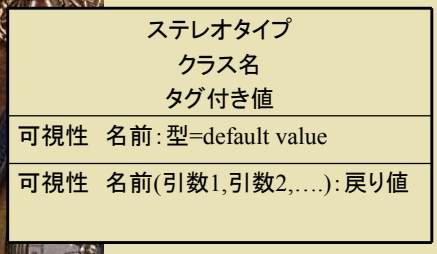
「１つ１つのユースケースの完了が、それぞれアクターの目的を満たす」

- アクター
  - 人間の形をしたスティックマンと、その下に役割を表すアクター名を表記する
- ユースケース
  - 楕円を書き、その中にシステムの機能を一言で表記する
- 関連
  - アクターとユースケースが作用する関係を線で結ぶ



クラス図

- クラス図
  - クラス と クラス同士の関連を表し、システムの静的分析に用いられる
- クラスの表記 （それぞれ名前以外は省略可）
  - クラス名 [パッケージ名::クラス名]で表記
    クラス名の上にステレオタイプ名、クラス名の下にタグ付き値を表記可能
  - 属性名 (property) [可視性 名前:型=default値]で表記
  - 操作名 (method) [可視性 名前(ﾊﾟﾗﾒｰﾀﾘｽﾄ):返り値]で表記



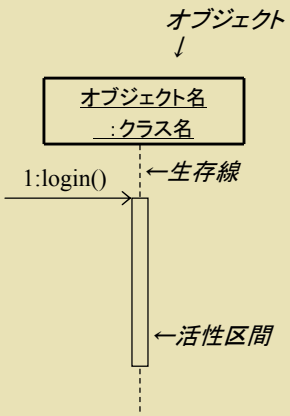
クラス名
属性名
操作名

可視性の表記記号について
+: public (どこからでも利用可)
~: package (同一パッケージ内で利用可)
#: protected
(同一パッケージ内と派生クラスから利用可)
-: private (クラス内でのみ利用可)

シーケンス図

- シーケンス図
  - オブジェクトの動的な振る舞いを時系列で表す
- オブジェクト（オブジェクト図参照）
  - オブジェクトを四角形で表記
- 生存線（ライフライン）
  - 破線で示し、オブジェクトがシステム内で生存している期間を表す。
  - 生存の終わりは単に破線が切れるか、明示的に×を示す
- 活性区間
  - 縦長の長方形で示し、オブジェクトがアクティブな期間を表す。（制御が移っている状態）
- メッセージ
  - 実線矢印で表現し、矢印の先のオブジェクトのメソッドを呼び出す。
- メッセージラベル
  - シーケンス番号：メソッド名（引数）

  ただし、メッセージがネストしている時は、
  「番号．番号．・・・」の形式にする。
  例　1.1.3 ログイン（　）



文字のみの設計書の例

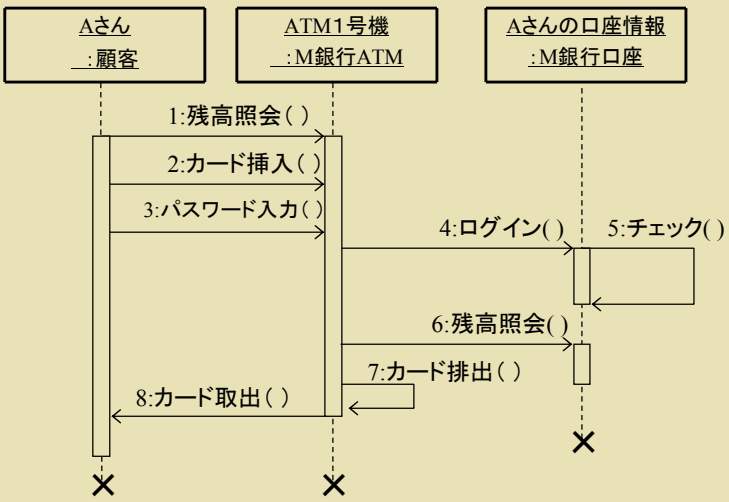
２．３．１　残高照会処理
本残高照会処理は次のように行う。

１．メニューで「残高照会」を押下する
２．キャッシュカードを挿入する
３．パスワード(暗証番号)を入力する
４．センターにあるホストコンピュータに接続して、
　　パスワードチェックを行う
５．残高を画面で表示する
６．カードを排出する

もし、途中でトラブルがあれば、カードを排出して
初期画面に戻る

シーケンス図の例

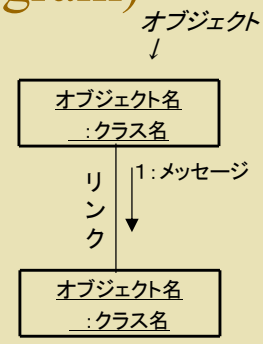
- AさんがM銀行のATM1号機で残高照会をします。残高照会にはカードと暗証番号入力が求められます。
- ATMは残高表示後にカードを排出して、処理を終えます。



コラボレーション図 (collaboration diagram)

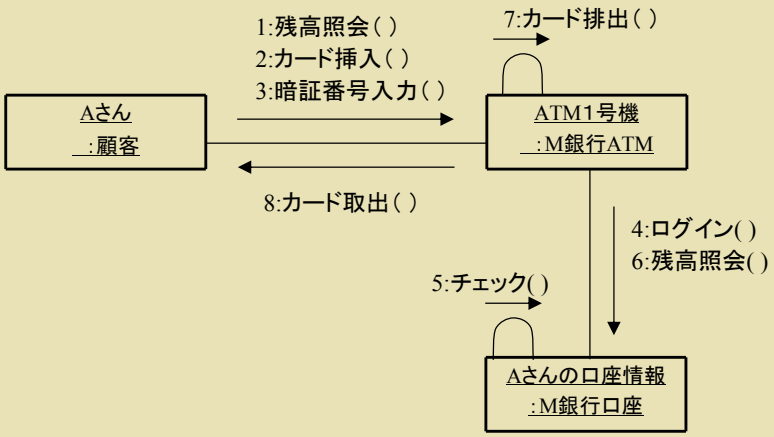
- コラボレーション図
  - オブジェクトの動的な振る舞いとオブジェクト間の関係を同時に表す
  - シーケンス図は時系列の順序を表現しており、コラボレーション図は空間的構造を表している。
- オブジェクト(オブジェクト図参照)
  - オブジェクトを四角形で表記
- リンク
  - オブジェクト同士を関連付け、クラス間の関係(アソシエーション)のインスタンスを表す
- メッセージ
  - 実線の塗潰し三角矢印で表現し、矢印の先のオブジェクトのメソッドを呼び出す
  - 必ず、シーケンス番号(順番)を付ける
- メッセージラベル
  - シーケンス番号:メソッド名(引数)

  ただし、メッセージがネストしている時は、
  「番号.番号.・・・」の形式にする。
  例 1.1.3 ログイン( )

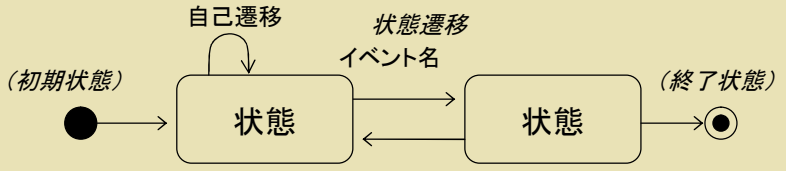


コラボレーション図の例

- AさんがM銀行のATM1号機で残高照会をします。残高照会にはカードと暗証番号入力が求められます。
- ATMは残高表示後にカードを排出して、処理を終えます。

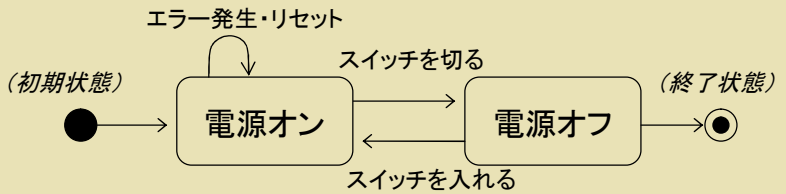


ステートチャート図
(State Chart Diagram)



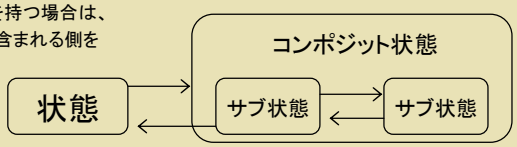
- ステートチャート図
  - 1つのクラスに着目し、そのクラスのオブジェクトが持つ状態の変化と、変化を引き起こすイベントを表したもの
- 状態
  - あるオブジェクトが置かれている状況を表す(待ち状態や、時間のかかる処理を実行中など)
  - 初期状態を●、終了状態を内側を塗りつぶした◎で表記
- イベント
  - オブジェクトが状態を変化させるきっかけの出来事

ステートチャート図



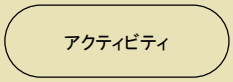
- 遷移： オブジェクトの状態がある状態からある状態へ移行すること
  - イベント
    - 「イベント名[ガード条件]／アクション」
  - ガード条件： 遷移が起こる条件
    - 例： スイッチを入れる[エラー発生]／リセット
  - アクション： 遷移が起こる前に実行される
- 状態の入れ子表記

ある状態の中に複数の状態を持つ場合は、含む側を、コンポジット状態、含まれる側をサブ状態として表現できる。

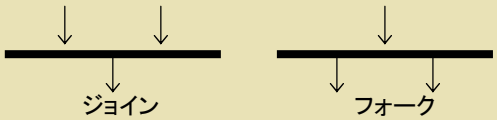


アクティビティ図(Activity Diagram)

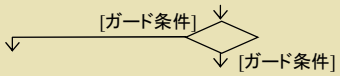
- アクティビティ図
  - 流れ図形式で、振る舞いの流れを表現する
- アクティビティ
  - オブジェクトが処理を行う状態



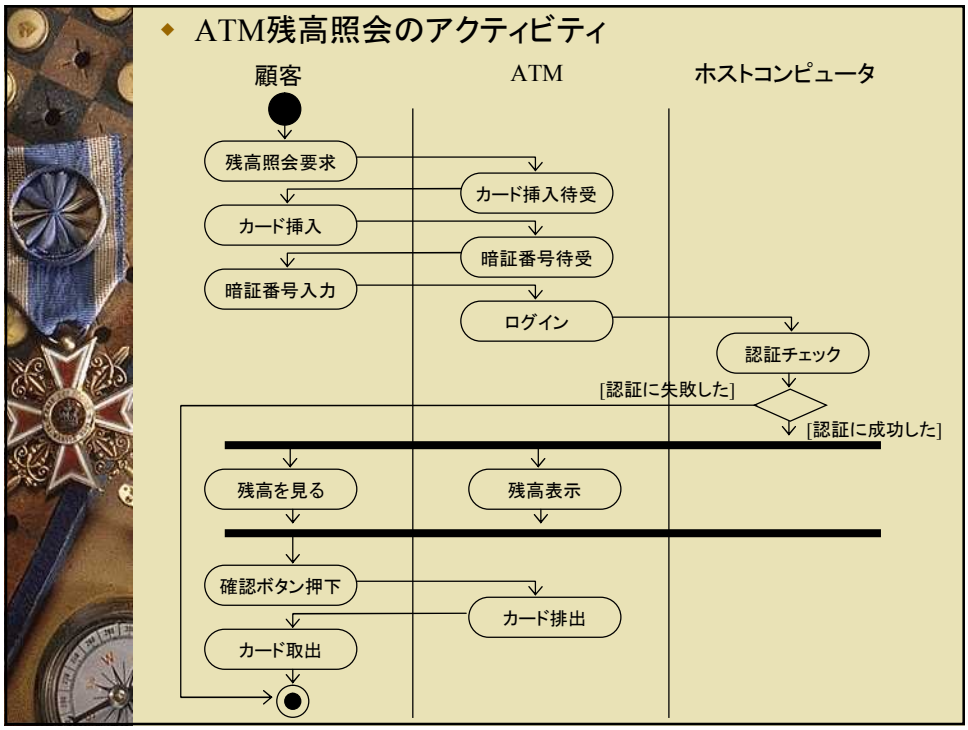
- 同期バー
  - 同時に行われる振る舞いを矢印と同期バーで表す



- 分岐
  - 条件[ガード条件]によって処理を分岐する



- レーン
  - 縦棒で区切られたアクティビティの状態


ATM残高照会のアクティビティ
顧客
ATM
ホストコンピュータ
残高照会要求
カード挿入待受
カード挿入
暗証番号待受
暗証番号入力
ログイン
認証チェック
[認証に失敗した]
[認証に成功した]
残高を見る
残高表示
確認ボタン押下
カード排出
カード取出