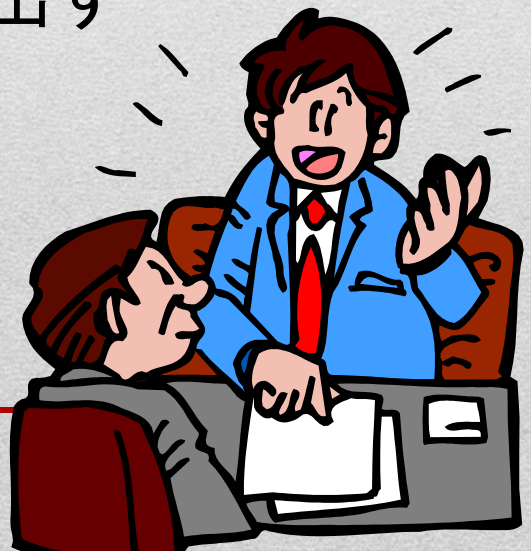
ソフトウェア工学

第13回 ユースケース図

ユースケース，ユースケース図

- ユースケース
  - 開発システムの機能要求の理解
  - 特定のジョブを完了するための開発システムの使用事例
  - 同じ目的をもつ，複数のシナリオを，一つにまとめたもの
  - 顧客との会話から開発システムとして必要なものを記述していく
- ユースケース図
  - 図示することにより，不要な機能を洗い出す
  - 開発システムが持つであろうUIと整合しなくてよい



シナリオ

- アクター と システム 間の一連のやり取り
- シナリオ 一つにつき起こり得る流れを記述
- 下の例だと，他に，クレジットカード認証に失敗した例のシナリオが考えられる.
- 複数のシナリオを用いて，一つのユースケースを記述
- シナリオの例（オンラインショップでの購入）:

顧客がカタログに目を通し，ショッピングバスケットに商品を入れます．支払いの際，顧客は配達方法とクレジットカード番号を入力し，購入を指示します．システムがクレジットカードを認証すると，直ちに購入内容を確認し，電子メールで連絡する.

アクター

- ロール の誤訳
- ユーザ または開発システムと相互作用を行う他のシステム
- アクターの名前（アクター名 ）は顧客とシステム設計者 の両方が理解できる名前をつける.
  - 顧客の業界 で使われている単語をアクター名につける
  - 顧客にとってなじみやすい
  - システム設計者が顧客の業界固有の概念になじみやすくなる
- アクターには，直接、開発システムを操作しない であろうユーザも必要に応じて記述
  - 監査担当者，インストール担当者，保守担当者，システムクロック

ユースケース (1/2)

- ユースケース名

  アクター を支援する手段 名

  （開発システムは）「YをZする 」という文章にする
- 概要
- アクター
- 目的

  ユースケースの存在意義を書く.

  「Zするため 」のように記述
- 事前条件

  ユースケースを実行する前の状態／条件 を書く

ユースケース (2/2)

- 事後条件

  ユースケースを実行された後の状態を書く

- 基本系列／基本フロー／主成功シナリオ

  アクターと開発システムのやり取りを書く

- 代替系列／代替フロー／拡張

  例外処理や条件分岐したときの処理（やり取り）を書く

その他として，備考欄，変更履歴，IDなどを書く

ユースケース の例

- ユースケース名　商品を販売する
- アクター　顧客
- 目的　クレジットカードを用いて販売するため
- 事前条件　顧客はシステムにログインしている
- 基本フロー

1. 顧客がカタログに目を通す.
2. ショッピングバスケットに商品を入れる
3. 支払いの際，顧客は配達方法と
   クレジットカード番号を入力し，
   購入を指示する.

ユースケース の例

4. システムがクレジットカードを認証する
5. 購入内容を確認する.
6. 電子メールで連絡する.

- 代替フロー

4a. 認証に失敗した場合

4. 顧客はクレジットカード入力を再入力する

ユースケースの補足

- シナリオは与えられない.
  - 顧客との打ち合わせ を通して記述していく
  - 顧客も設計者も開発システムがない 状態で打ち合わせをするので，シナリオの形に落とす ことが設計の第一歩となる
- 顧客の期待や原要求 をユースケースでは記述しない
  - 開発システムの設計に落ちるような機能と とその手順を記述
- ユーザインターフェース の設計をユースケースでは記述しない
  - 他のモデルを使うべき
- 基本フロー など細かい手順は相手によっては書かない
- ユースケースとはアクターに対して何らかの結果を 返すものである.

ユースケースを書いてみよう

- シナリオ

学生が端末に学生証をかざすと，端末は学生証から学籍番号を読み取り，当該学生の学割発行回数をチェックする．一日2枚まで，もしくは，一年に15枚までという上限に達していなければ，学割を発行する．

ユースケースを書いてみよう

・ユースケース名　学割を発行する
・アクター　学生
・目的　端末で学割を発行するため
・事前条件　学生が学生証を持っている

・基本フロー
　１.学生が学生証をかざす
　２.学籍番号を読み取る
　３.学割発行回数をチェックする
　４.上限以内なら学割を発行する

・代替フロー
　2b.学籍番号が読み取れない場合
　　４.１をお願いする

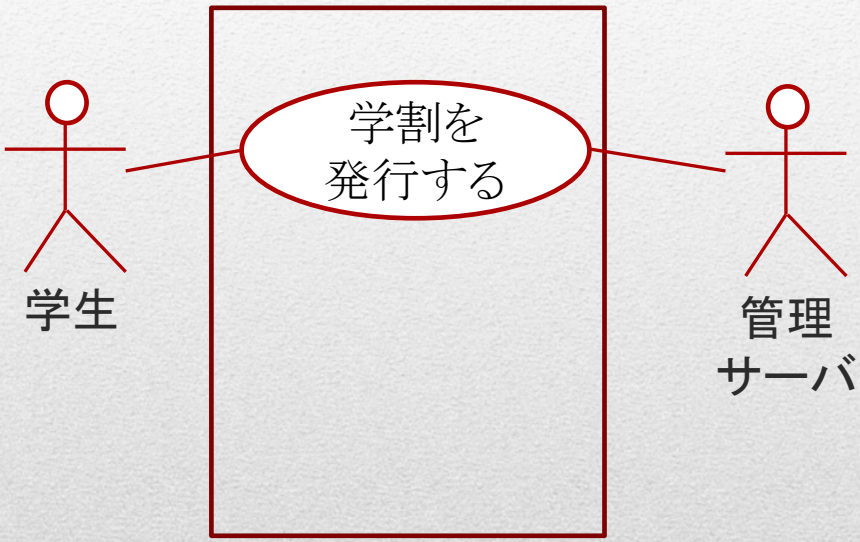
　4b.上限を超えていた場合
　　４.上限を超えていますと表示する

ユースケース図

- ユースケース
- アクタ

  他のアイコンであっても良いが，
  開発システムに影響を与えるものが
  外部システム であっても人型の
  アイコンを用いることが多い
- 境界枠

  開発システムと外部 との境界
- 関連



ユースケース間，アクタ間の関係

- 汎化（Javaでいうところ継承）

拡張の概念

- <<include>> 関係

包含される ユースケース

ユースケース間の 共通処理 を

図示することができる

- <<extend>> 関係

Javaでいうところの 継承ではない

拡張して付け加えたユースケースを図示