

# 学生向けモデリング演習 支援システムの開発と評価

---

2011.03.18

第109回コンピュータと教育研究発表

静岡大学 情報学研究科

小木曾禎 遠山紗矢香 湯浦克彦

# 研究の背景

- 2009年にTAを勤めた静岡大学「情報システムデザイン演習」におけるモデリング演習にて、演習を円滑に行えない学生が存在することに気が付いた
- モデリングはシステム開発において重要な位置を占める
- モデリング教育は情報系の高等教育機関に求められる期待の大きな科目
  - ⇒しかし、モデリングの目的、実施方法や良いモデルへの判断基準などは単純に言い表しにくいものであり、初学者にとっては学習に困難を感じやすい科目

# 研究の目的

- 目標

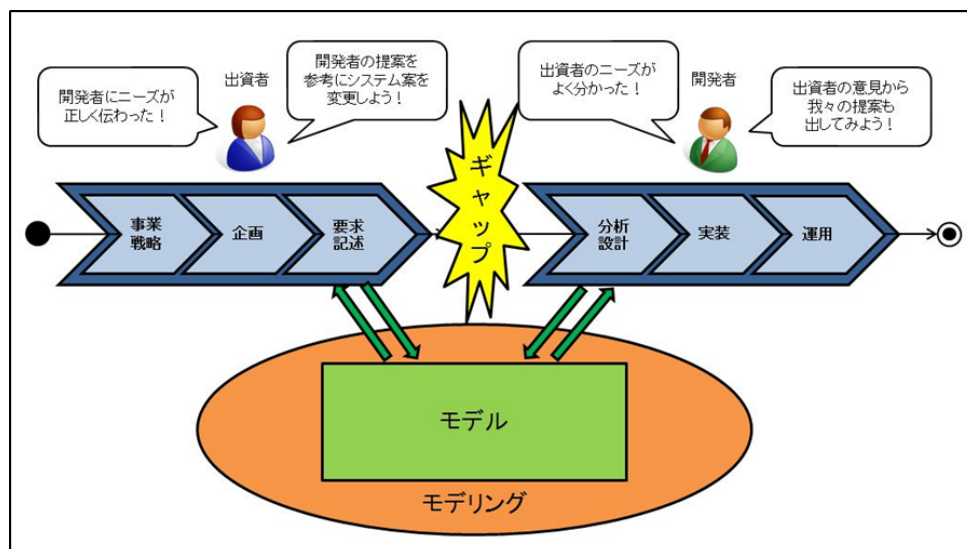
- 情報系の高等教育機関のモデリング教育において、モデリング演習を円滑に行えない学生をなくすことを最終的な目標としている

- 目的

- 一事例として静岡大学のモデリング演習を受講する学生が理解に行き詰る根本原因を調査する
- 調査に基づいてモデリング演習を支援するシステムを開発し、使用させて評価を行う

# モデリング

- システムの目的や構造を、ある範囲の記述形式で概要を記すこと  
⇒ 視覚的に理解しやすいモデルを作成すること
- システム開発の出資者と開発者（または、上流開発者と下流開発者）の意識のギャップを埋めるコミュニケーションツール



(児玉公信著「モデリングの本質」(日経BP社、2004)、p.12を基に編集)

# モデリング教育の難しさ

- システムの要求や機能は実体がないので表しにくい
- 学生によって意欲に差が出る
  - 多くの学生は意欲的
  - 一方で、要領が分からず意欲を低下させていく学生も存在
- 全員への必修科目として実施
  - 学生の前提知識が文理で異なる
  - 学生によって進捗度が異なる
- コミュニケーション力が必要となる

# 先行研究及び先行事例

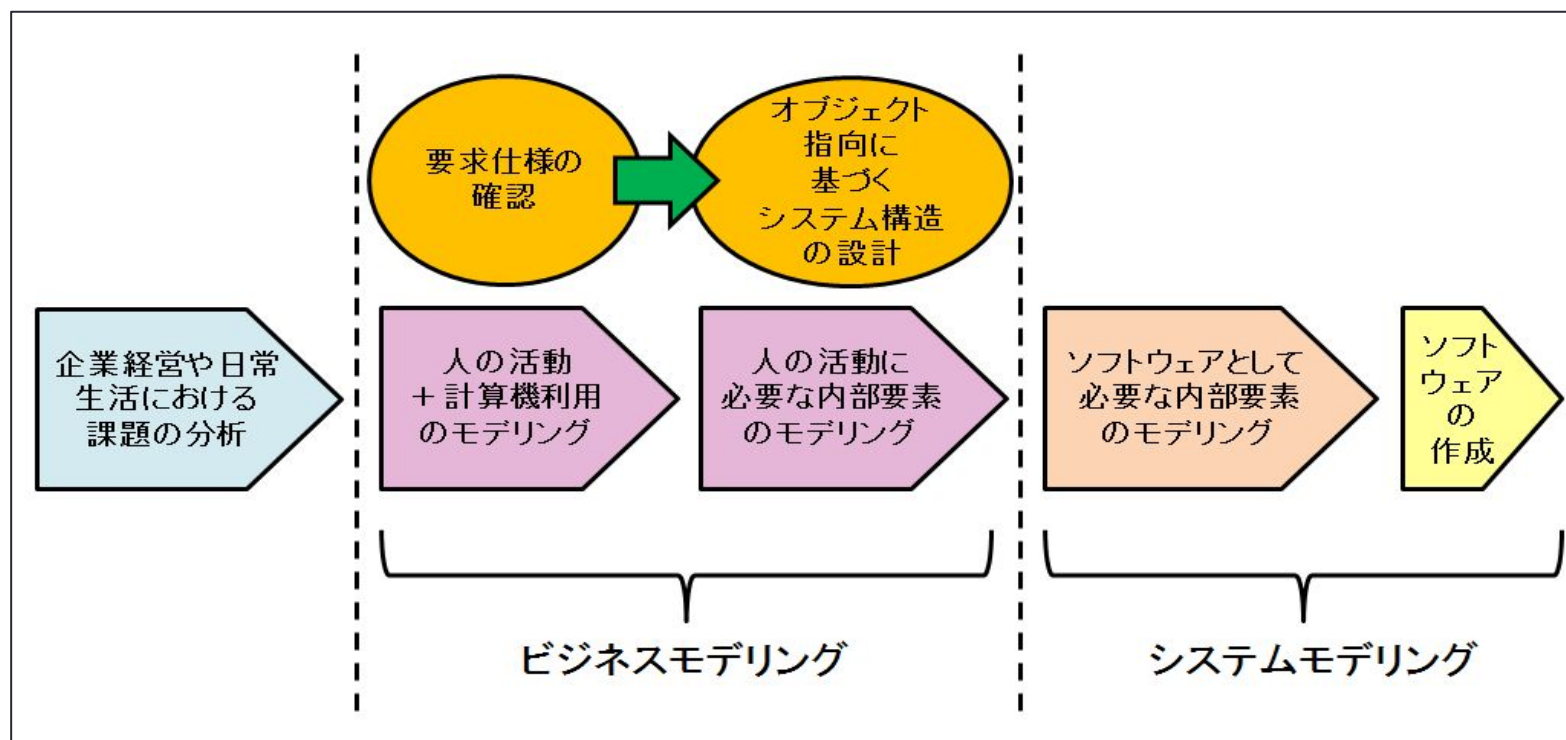
- 演習におけるコミュニケーション支援
  - 設計演習のための協調作業支援ツールの開発
    - 東京学芸大学のGPSS（プログラミングに関わる設計におけるグループでの作業管理や成果物の作成・登録を支援するツール）
    - 静岡大学の協調学習支援システム（グループを単位とする議論や成果物作成・提出や教員との応答などの機能を含むシステム）
  - 汎用的なコミュニケーション支援ツールの利用
    - Moodle、Blackboardなど
- 設計に関する知識の提供
  - UML（ Unified Modeling Language ）に関する多くの教科書
  - 教員による欠陥の指摘
    - 東京学芸大学（指摘情報を学生が参照できる環境を提供）

# 静岡大学「情報システムデザイン演習」におけるモデリング演習

- 対象 IS(情報システム)プログラム2年生約80名
  - 理系入試約30名＋文系入試約50名
  - 1クラス約40名(約4名×10グループ)
- 時間 週2コマ×7週
- 目的
  - オブジェクト指向に基づくビジネスモデリングを演習により理解する
  - グループ内の合意形成や外部への分かりやすい説明など、システム開発に求められる行動特性を理解する

# 演習の範囲としてのビジネスモデリング

- オブジェクト指向モデリングの2つの範囲
  - ビジネスモデリング(本演習の範囲)
    - 要求仕様の確認とシステム構造の設計を行う
  - システムモデリング
    - プログラミングの仕様となる詳細な設計を行う

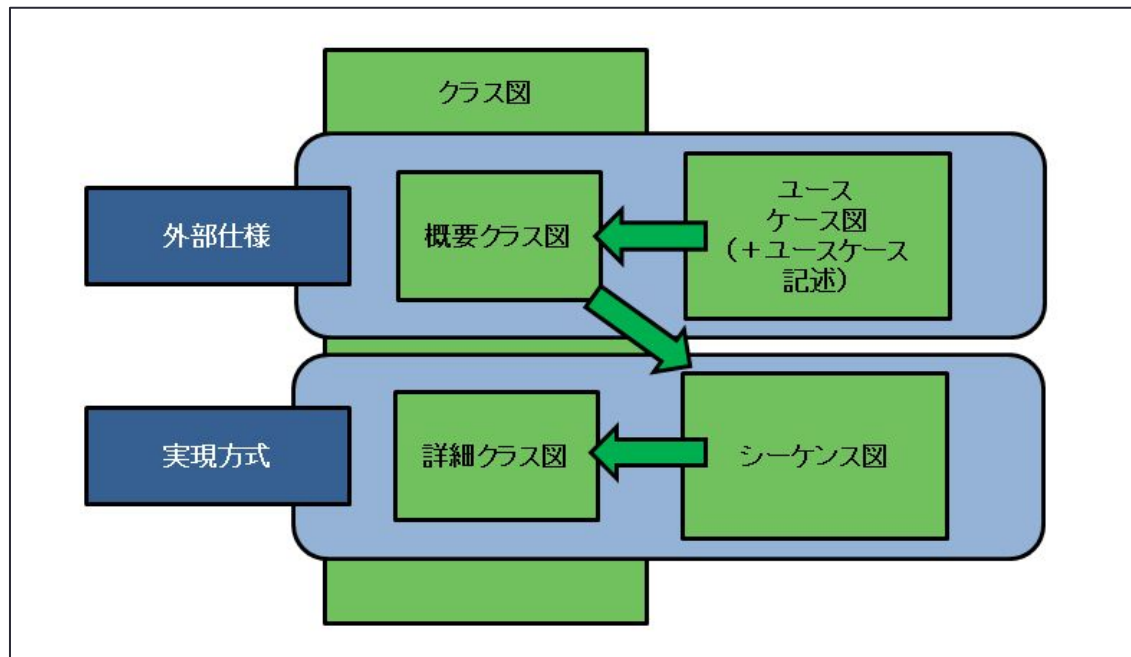






# モデリング演習の手順(2)

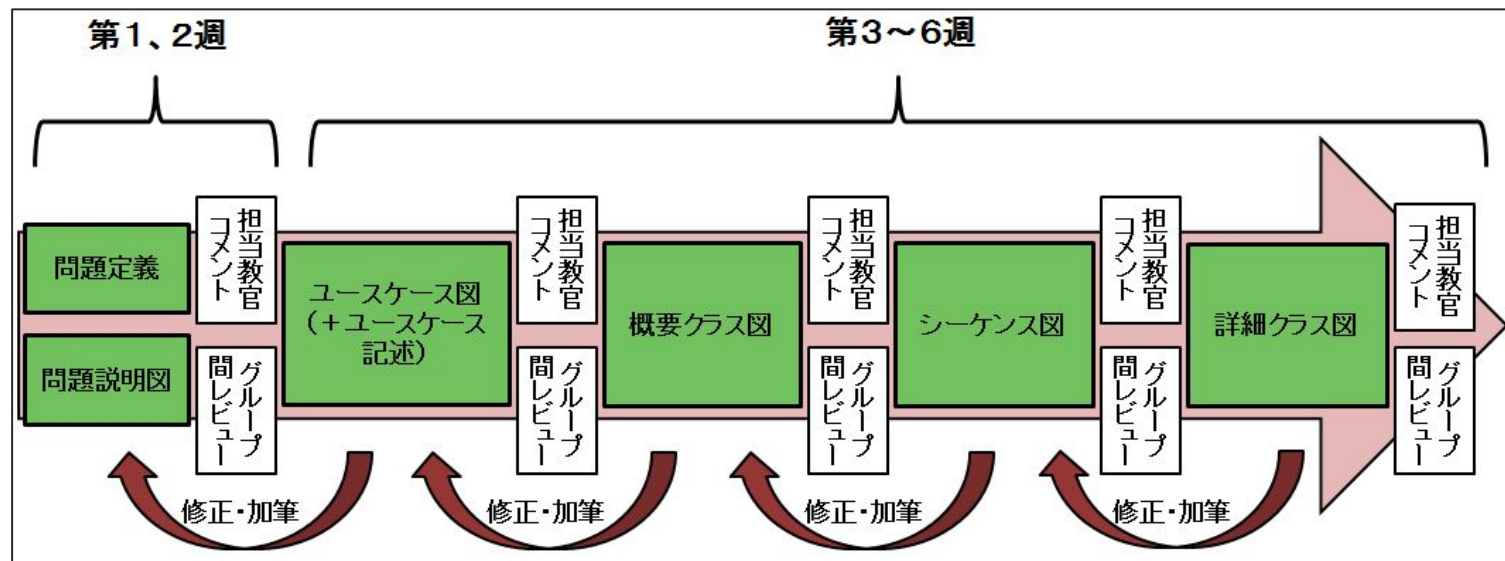
- 第3週～第6週:「オブジェクト指向に基づくシステム構造の設計」演習
  - システムの外部仕様と実現方式の設計
  - ユースケース図(+ユースケース記述)、クラス図(概要、詳細)、シーケンス図の作成
  - astah community(UML描画ツール)の使用



- 第7週: 成果発表会

# モデリング演習におけるコミュニケーション

- グループ内での議論と合意形成
- グループ間での説明とレビュー
- 担当教官とのやり取り
- 発表会





# 問題の原因分類

- 問題を7つの原因分類でまとめた

## 原因分類

①モデリング演習の目標  
やモデル作成アプローチ  
に関する  
説明不足・理解不足

②モデルの設計ノウハウ  
が分からない

③UMLに関する  
用語・使用方法  
が分からない

④astah communityの  
使用方法の説明不足

⑤受講生の  
モチベーションが  
高くない

⑥グループワークで  
協力できない

⑦ハードウェアの故障

# 原因への対策案の検討

## 原因分類

①モデリング演習の目標  
やモデル作成アプローチ  
に関する  
説明不足・理解不足

## 対策案候補

A: 講義の中で  
説明を繰り返す

B: 個別に  
説明を繰り返す

C: 受講生に質問  
させる

D: 受講生同士で  
教え合う

E: モデリング演習  
教材・コンテンツの  
提供

## 対策案候補の検討

・時間不足  
・効果のない学生がいる

・教員不足

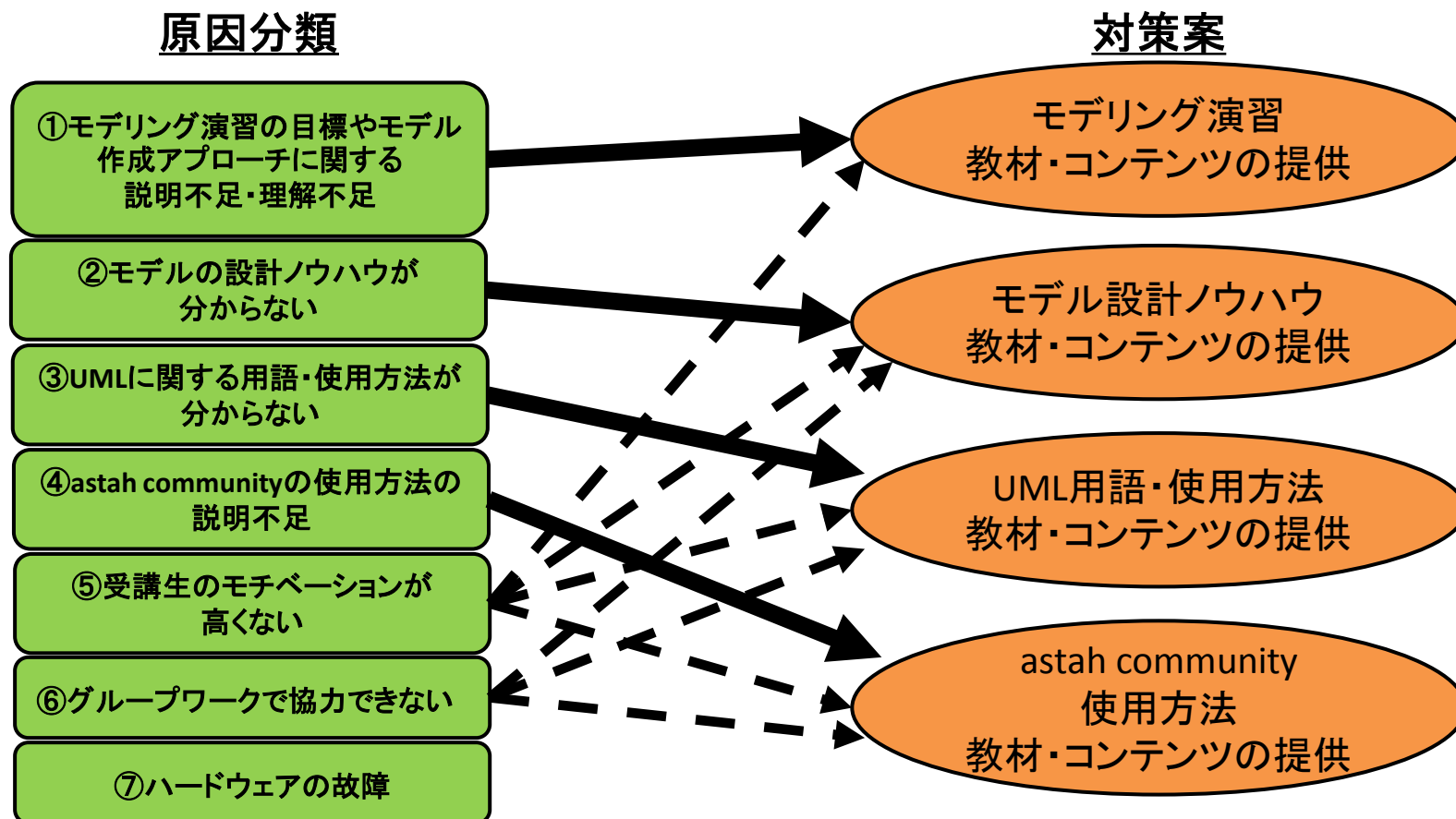
・教員不足  
・できない学生がいる

・既に取り組み済み  
・できない学生がいる

・どんな学生でも対応可能  
・学生が自主的に学習可能

↑  
対策案  
として重点化する

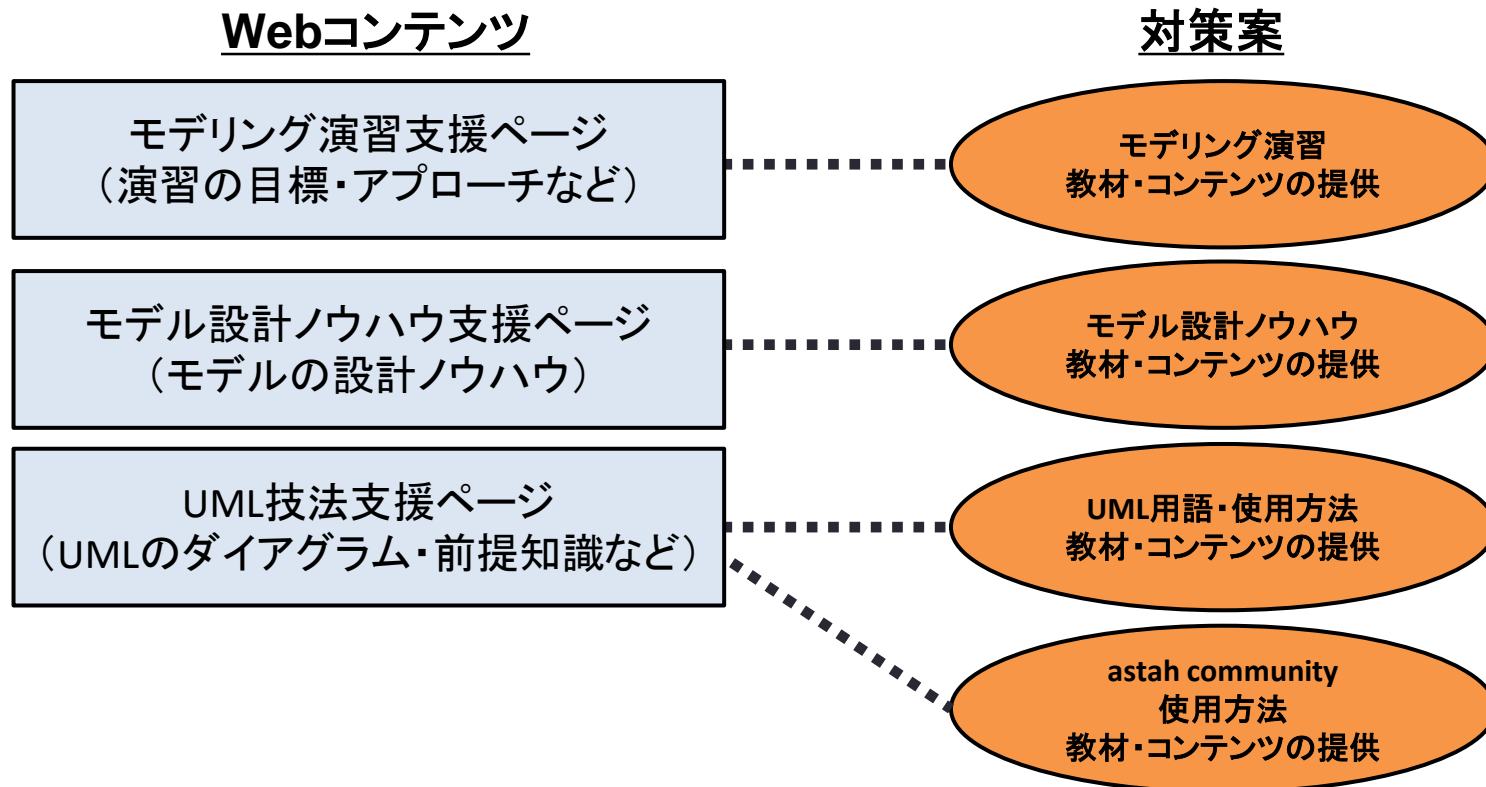
# 対策案のまとめ



- 太い実線は解決できる期待が大きいことを示し、細い波線は解決できる期待が太い実線よりは小さいことを示す

# モデリング演習支援システムの開発

- 対策案はWebコンテンツとしてシステム化し、受講生に提供
- 対策案はそれぞれ次の3つのWebコンテンツとして実現





# モデリング演習支援ページ

- モデリング演習の目標、モデル作成へのアプローチ、演習手順について解説する(授業の説明を補う)

- 1.モデリングとは？
- 2.要求仕様の確認演習 → 業務雛形の提供
- 3.オブジェクト指向に基づくシステム構造の設計演習
  - 3.1.オブジェクト指向に基づくシステム構造の設計演習とは？
  - 3.2.ユースケース図とユースケース記述の作成
  - 3.3.概要クラス図の作成
  - 3.4.シーケンス図の作成
  - 3.5.詳細クラス図の作成

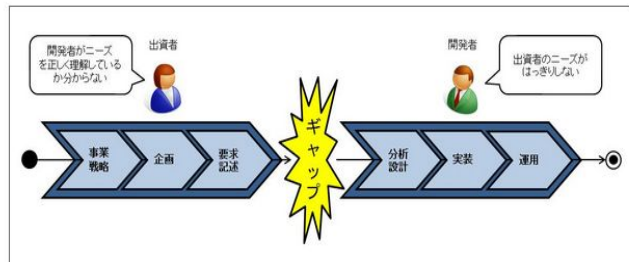
# モデリング演習支援ページ 工夫点A、B

## モデリングとは？

例を挙げます。ある出資者がある開発者に新しいシステムの開発を依頼するとします。出資者は事業戦略に基づき、新しく必要とするシステムの企画を行い、開発者にどのようなシステムが欲しいのかを伝えるために要求記述を作成しました。そして開発者は要求記述から分析と設計を行い、実装と運用を行います。

しかし開発者が完成したシステムを納品したとき、出資者は「こんなシステムが欲しいのではなかった…。」と不満を述べました。

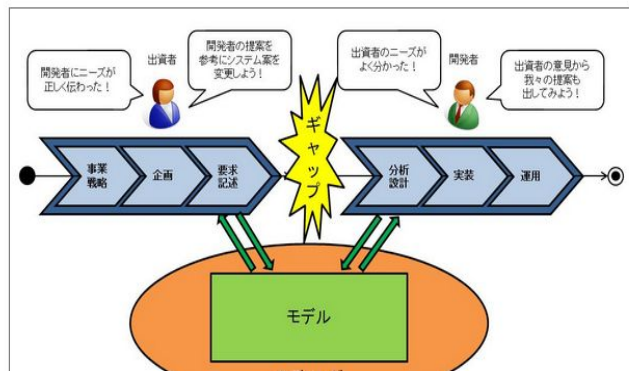
このような事態が発生した最も大きな原因として、出資者の要求と開発者の解釈の間にギャップが発生したことが考えられます(図m1-1)。



●図m1-1 出資者の要求と開発者の間のギャップ(見玉公信著「モデリングの本質」(日経BP社、2004)[14]、p.12を基に編集)

そこでこのギャップを埋めるために考えられたのが、出資者と開発者が互いに理解できるようなモデルを作ることです。このモデルを作ることをモデリングと呼びます。

モデリングの利点は、出資者の要求を正しく開発者に伝えるだけでなく、システム制作のプロである開発者からの提案も出資者に伝えることができるようになります。より確立したシステムを開発できるようになります(図m1-2)。



モデリング演習の必要性から目標を説明



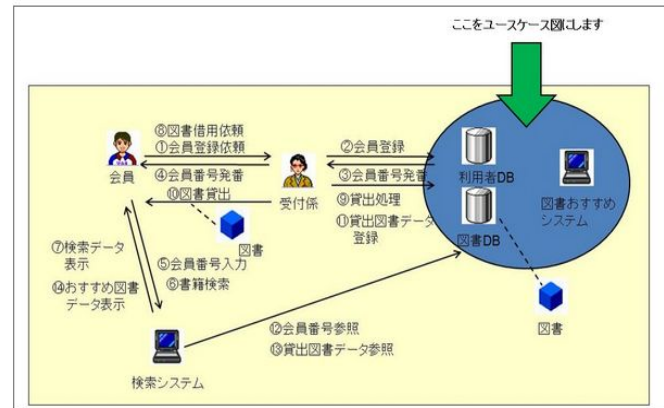
実例を挙げ、ステップに分けたモデル作成アプローチの説明

## ユースケース図の作成

問題定義と問題説明図からユースケース図を作成します。アクターとアクターに提供する機能となるユースケースを抽出します。

アクターやユースケースについてはUML技法支援ページ第7章を参考してください。

先ほど例として問題定義と問題説明図の作成で設定した「図書おすすめシステム」のユースケース図を作成します(図m3-2)。



●図m3-2 ユースケース図を作成する部分

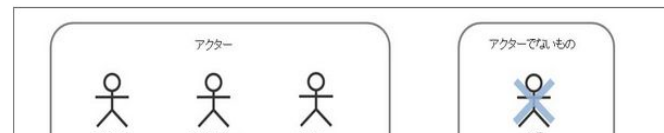
ユースケース図は提案するシステムによって描く必要になる数も変わり、提案するシステムが増えればその分増やすこととなります。

しかし、1つのシステムに関するユースケース図をいたずらに分けて記述することは推奨されません。事例モデル評価支援ページ(事例⑧)「学務情報システム」では、ユースケース図を4つに分けているため、システムの機能が分かりにくくなっています。

## 〈ユースケース図手順1)アクター

アクターはシステムの利用者です。「図書おすすめシステム」はすべて図書館の受付係が操作します。会員登録も会員が自ら行うのではなく、図書館で受付係が行います。従って受付係はアクターですが、会員登録しようとする人や会員はアクターではありません。つまりアクターとはシステムと直接対話をする人です。

人以外にもこのシステムと直接データをやりとりする外部のシステムもアクターです。今回既存の利用者DB(システム)と図書DB(システム)と連携するので、利用者DBと図書DBもアクターです(図m3-3)。



# モデリング演習支援ページ 工夫点C、D

**業務雛形の提供**

下に4分類12種の業務の雛形を提供します。この中(①~⑫)から新システムを提案する業務を選択しましょう。また、選択するのは1つである必要はありません。2つ以上を組み合わせ、複合型の業務として試してみてもかまいません。

業務雛形分類1:メッセージの処理

<共通機能>  
依頼者、受理者、処理者の間で、メッセージによって処理の依頼と返答を行う。受託者の仕事は、大部分が情報システムによって自動化される。処理口は、情報システムによるものと、人間が行うものがある。

●図11 メッセージの処理の業務相関図

①業務雛形分類1-1:レストランの受付  
<機能>  
メッセージの処理共通機能に加えて、  
・料理を配送する。  
・客より作成状況問い合わせ(催促)を受け付け、回答する。  
・その他、メニューの提示、メニュー項目の説明など。

●図12 レストランの受付の業務相関図

工夫点C

業務雛形の提供・解説

工夫点D

3層モデルを想定したクラスの分析手法の紹介

〈シーケンス図手順1〉3層モデル化

シーケンス図は基本の単位としてユースケースごとに作成します。先の例の「図書を出し出す」のユースケースからシーケンス図を作成しましょう。このユースケース記述では、シーケンス図を作成する前に3層モデル化を行うと良い図が作成できます。

3つのステレオタイプ << boundary >>、<< control >> および << entity >> を用いてユースケースを分析することを3層モデル化と呼びます。  
では順番を追ってユースケース「図書を出し出す」を3層モデル化します。

(1)バウンダリクラス  
バウンダリ(boundary)の意味は境界です。バウンダリクラスはシステムの外部にいる **アクター**と直接対話をするための **クラス**です。アクターが人なら画面クラスです。アクターが外部システムならその外部システムとインタフェースをとるクラスです。  
バウンダリクラスはそのユースケースのアクター毎に1つ作ります。クラス名は「アクターが人なら「ユースケース名(名詞) + 画面」、アクターが外部システムや外部の機器なら「システムまたは機器の名前 + インタフェース」とします。今回の例では、ユースケース名を名詞型に短縮して「図書貸出画面」とします。  
画面のレイアウトや画面遷移などはここではまだ考えません。m3-19以下3枚の図は理解を助けるためのイメージです。

●図m3-19 バウンダリクラスを追加する

(2)コントロールクラス  
コントロールクラスはその言葉の意味通りユースケースをコントロールするためのクラスです。ユースケース毎に1つ作ります。  
クラス名は「ユースケース名(名詞) + コントローラ」とします。今回の例では、ユースケース名を名詞型に短縮して「図書貸出コントローラ」とします(図m3-20)。

# モデル設計ノウハウ支援ページ

- 学生が悩みやすい設計ノウハウを解説する(過去の学生が提出したモデルを事例とする)

- (事例1)「CDのレンタル」...「人の行動」と「システムの機能(動作)」を区別し、かつ一貫させる事例
- (事例2)「旅行計画」...案件とそれを扱う構造の記述方法に慣れる事例
- (事例3)「学務情報システム」...拡張を続ける設計活動に対応する事例
- (事例4)「ピザの宅配システム」...ビジネスモデリングとシステムモデリングを使い分ける事例

# モデル設計ノウハウ支援ページ 工夫点A、B

ユースケース図とユースケース記述

■成果物(図s1-5~図s1-11)

## ユースケース図



●図s1-5 ユースケース図

## ユースケース記述 <顧客を登録する>

1. 顧客が管理者に対して登録を依頼する
2. 管理者は顧客から個人情報をうけとる
3. 管理者は顧客の個人情報を顧客情報として顧客DBに、登録する
4. システムは顧客ごとに顧客IDを割り振り、会員証を発行する
5. 管理者は顧客に対し、顧客IDの載った会員証を渡す
6. 顧客が管理者に対して退会を依頼する
7. 管理者は顧客情報を削除する

工夫点A

過去モデリング演習  
受講生作成モデルを  
事例として提供

工夫点B

事例の指摘や評価を  
学生に分かりやすく  
解説

## ■評価

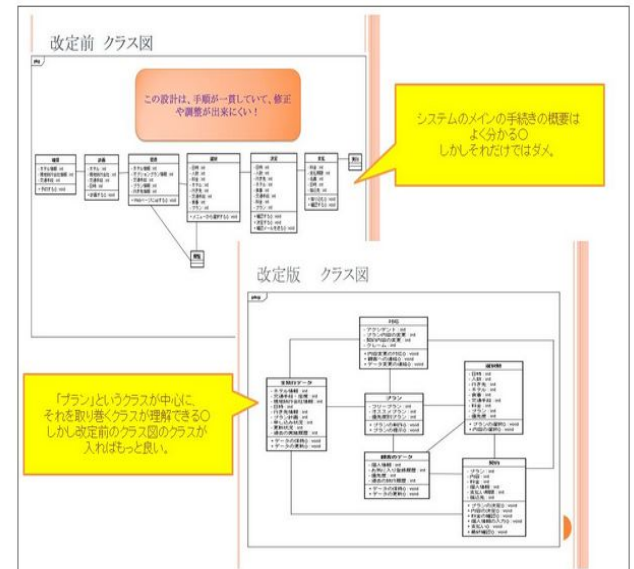
### ☆手順で分けるか、構造で分けるか

・システムを用いる仕事は、第一に手順として把握することができます。そこで、システムの構造(クラス)も手順に沿って分割するのが一つの考え方ですが、それだけではあまり上手な設計とは言えません。各手順に共通した対象物の見方や処理の方法に着目して分割するのがオブジェクト指向の主旨とするところですので、もう一度対象を深く掘り下げて見る必要があります。

・手順から構造へと分割の視点を移すことによって、クラス図がガラリと変わったという例を紹介します。

改定前版のクラス図は、ユースケース記述「プランを表示する」に示した、いわゆるシステムのメインの手続きを順に進めることを念頭に、各ステップで用いる情報を示したものとします。各ステップの概要を知るうえでわかりやすい記述ですが、それだけであまりさすぎではありませんか？

さて、改訂版のクラス図では、プランというクラスを中心に、プランの要素となる旅行データとプランに基づく顧客の選択肢の関連が定義され、それらの情報をもとに顧客との契約が進むことやプランの変更などの対応がなされることが示されています。ユースケース記述のステップごとで用いる情報は、ここに書かれたクラスにまたがる部分もありますが、全体のクラスの定義(本質的な情報の関連)が理解できれば、馴染みあったクラス群と関連を持つこととなります。改定前版のクラス図のクラスとの対応も付けてみてください。容易にできるはずですよ(図s2-17)。



●図s2-17 改定前版のクラス図は手順、改訂版のクラス図は構造で表されている

# UML技法支援ページ

- UMLに関する用語・使用方法を解説する
- astah communityの使用方法を解説する

- (初級編1) 第1章UMLとは何だろう？
- (初級編2) 第2章モデルって何？
- (初級編3) 第3章UMLのオブジェクトを理解しよう
- (初級編4) 第4章同じ種類のオブジェクトをクラスにまとめる
- (中級編1) 第5章一番大切なクラス図を理解しよう
- (中級編2) 第6章シーケンス図を理解しよう
- (中級編3) 第7章ユースケース図を理解しよう
- (上級編1) 第8章クラスを分解して整理しよう
- (上級編2) 第9章多態性って何？
- (上級編3) 第10章オブジェクトを分解して内部構造を理解しよう
- (番外編1) 第11章その他

# UML技法支援ページ 工夫点A

- 工夫点A 対象の学生向けである、内容が分かりやすい教科書の選定
  - UML技法支援ページのコンテンツの内容に求められること
    - 講義を受講する学生向けであること
      - UMLのダイアグラム(ユースケース図、クラス図、シーケンス図)
      - オブジェクト指向の考え方に必要な知識
    - 内容が学生に対して分かりやすいこと
      - 初学者でも読みやすい
      - 初学者でも理解できる
- 河合昭男著「UML超入門-はじめてのモデリング-」(技術評論社、2010)の内容を基にコンテンツ化

# UML技法支援ページ 工夫点B

(7-4)astahを使ってユースケース図を描いてみよう

UMLツールを使ってユースケース図を描いてみましょう。

(7-8-1)ユースケース図を作成しよう

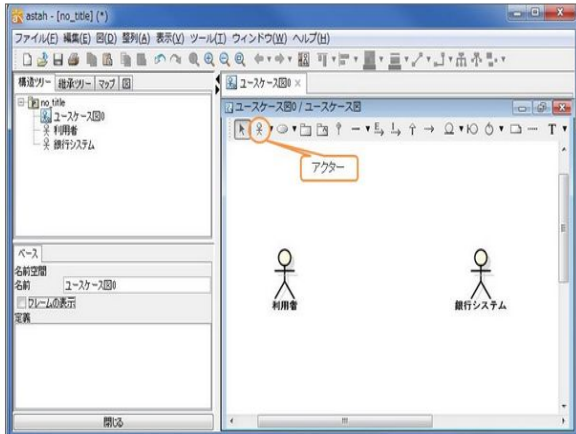
7-1節で示した銀行のATMのシナリオでユースケース図を作成しましょう。利用者は銀行のATMを利用して現金の引き出し、入金、他の口座に送金を行います。

このシナリオでは、アクターとして「利用者」と「銀行システム」、ユースケースとして「現金を引き出す」と「預金する」と「振り込む」が考えられます。(シナリオから「銀行システム」という単語は出てきませんが、アクターは人間の他、連携の必要な外部のシステムや機械の場合もあるので、シナリオにおける銀行のATMのシステムを「銀行システム」としています。もちろん伝わるのであれば、他の表現を用いてもかまいません。)

まずはastahを起動した後、左上の「図」から「ユースケース図(U)」を選択します。

選択したユースケース図上側に並べられているアイコンの左寄りにあるのがアクターアイコンです。クリックしてユースケース図内でクリックするとアクターが表示されます(図7-5)。

中央にユースケースを置くスペースを確保し、左側に利用者、右側に銀行システムを表示します。(アドバイスとして、複数のアクターが存在する場合、左側はユースケースを開始するアクター、右側はサービスを提供するアクターを表示します。)



●図7-5 アクターを表示する

アクターアイコン右側のユースケースアイコンを選択します。クリックしてユースケース図内でクリックするとユースケースが表示されます。

次に、ユースケースアイコンよりさらに右側の関連アイコンを選択し、アクターとユースケース間をドラッグすると関連付けを表示できます(図7-6)。

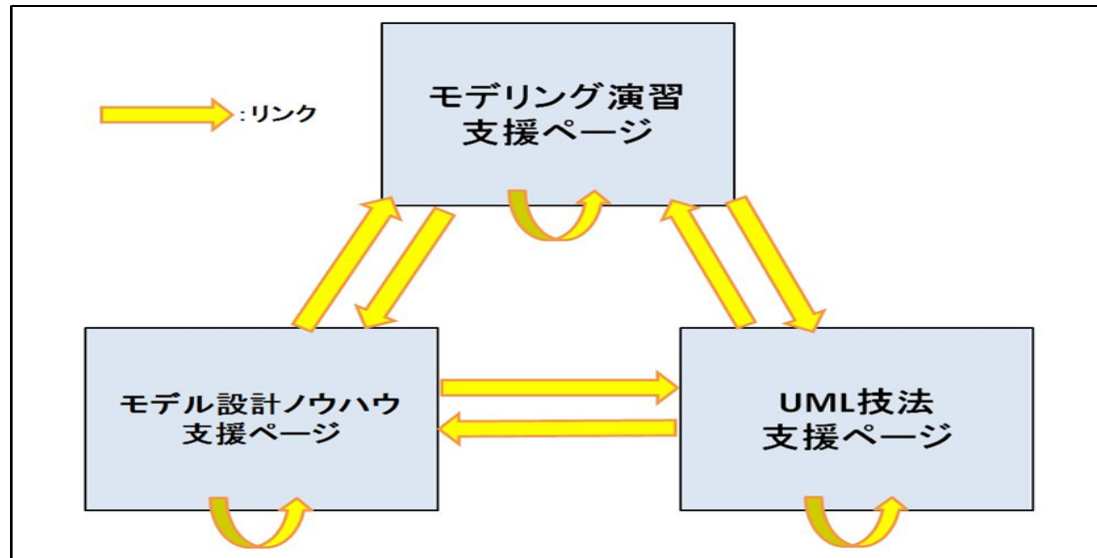


クラス図、シーケンス図、ユースケース図の説明に、astah communityによる描画方法の説明の記述



## 3つのコンテンツを相互に呼びだすリンクの設定

- 学生が演習を行う際に、いつでも必要に応じて他のコンテンツを参照できる
  - 「モデリング演習支援ページ」で演習手順や注意事項を確認しているときに、設計ノウハウを知りたくなった  
⇒「モデル設計ノウハウ支援ページ」へのリンク
  - 「モデル設計ノウハウ支援ページ」で設計ノウハウを確認しているときに、UML用語の意味を知りたくなった  
⇒「UML技法支援ページ」へのリンク



# モデリング演習支援システム構築方法

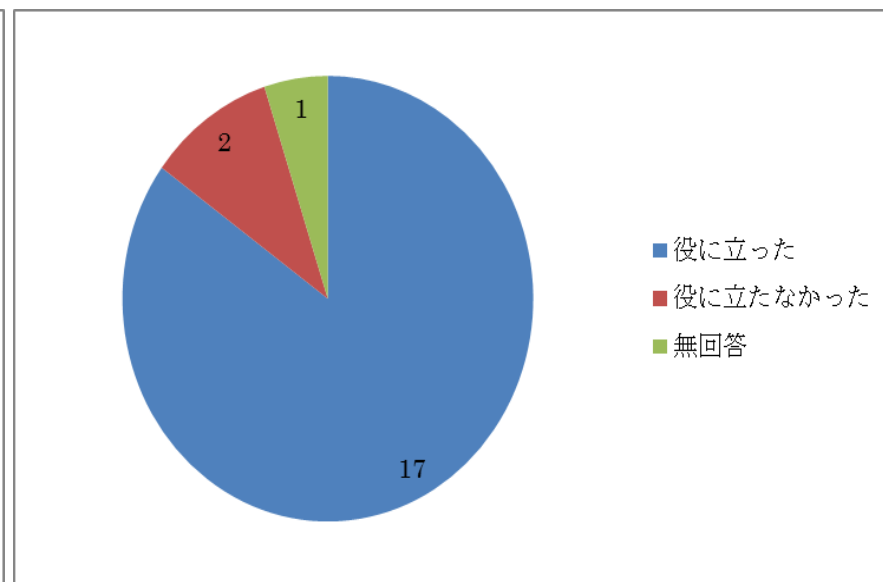
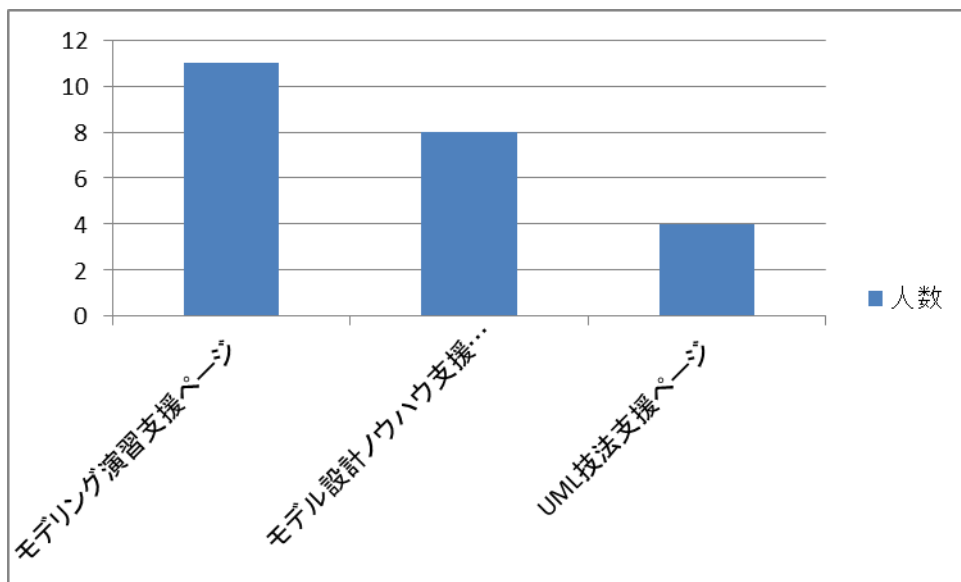
- モデリング演習支援システムはWebサイトとして構築した
- Webサイトの構築に利用したモノ
  - Webサイトを公開するホスティングサービス
    - ⇒Google App Engine
  - Webサイトの見やすさを向上
    - ⇒Design room Kannaが提供するcssデザインテンプレート

# 支援システムの公開

- 2010年度モデリング演習の後半クラスを受講する学生に公開
  - 担当教官に協力してもらい、講義の中でコンテンツ内容を紹介し、受講する学生の全員が一度は目を通してもらえるようにした
- 授業中の学生の様子を観察
  - モデリング演習支援ページ、モデル設計ノウハウ支援ページ、UML技法支援ページの順に多く利用されている様子であった

# アンケート結果

- 公開を行った受講生に対して、モデリング演習支援システムについてアンケートを行った(回答者は20名)
- 3つのコンテンツについて、半数以上の受講生がモデリング演習支援ページが有用であったと回答し、総合的な評価について、85%の学生が有用であったと回答した



# 学生からの感想

- 学生から役立ったという感想を頂いた
  - 「モデリングの定義から、シーケンス図の説明までとても丁寧にまとめてくれたWebページだったので、これからも参考にしたいなと思っています。」
  - 「2年前期のモデリングという授業で、ユースケース図やクラス図の作成方法を勉強したことがあるが、それらについてはあまり理解できなくて、うまく作成することもできていなかった。今回は先輩がモデリング演習支援システムを作ってくれたため、ユースケース図やクラス図の作成することだけではなく、システム設計に関して、他の詳しい知識も学ぶことができて、よかったと思う。」

# 結論

- ティーチングアシスタントとして情報システムデザイン演習のモデリング演習を観察し、受講生達がモデリング演習を行う上で障害となる問題の根本原因を分析し、これを解決する対策案を考察した
- そして、3つのWebコンテンツ(モデリング演習支援ページ、設計ノウハウ支援ページ、UML技法支援ページ)からなるモデリング演習支援システムを開発した
- 開発したシステムを講義において公開したところ、モデリング演習支援ページを中心に多くの受講生に使用され、総合評価として85%の学生から役に立ったという評価を得ることができた

# 今後の課題(1)

- モデリング演習という学習が困難な科目の授業に、補助となるコンテンツを提供するという試みだけは行うことができた

しかし

- 学生が円滑にモデリング演習を行えないことを解消することを目的に開発されたが、本当に解消されたかどうかは十分検証できなかった

# 今後の課題(2)

- 評価方法の見直し

⇒Webコンテンツが学生の理解を助けることができているかを調査する必要がある

- コンテンツ項目ごと
- 学生のタイプごと



この評価を基に

- モデリング演習支援システムの見直し

- 学生の参画度の向上



# 今後の課題(3)

- 学生の参画度の向上

参画度の向上

- コンテンツの閲覧

単純なグループ学習

- 他グループ成果へのコメントの提出
- レビュー議事録

現状

- 他グループからのコメントに基づく成果の改善
- グループ成果のランキング評価
- 教員からのクイズ

次の取り組み

- 学生が設計成果をリアルタイムで相互に議論
- 学生がコンテンツ(モデル設計ノウハウ支援ページ)を作成

最終目標