

ソフトウェア工学 第3回

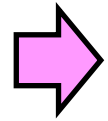
— 要求分析 —

大連理工大学・立命館大学 国際情報ソフトウェア学部

大森 隆行

講義内容

■ 要求分析



■ 概説

■ 要求獲得

■ 要求仕様化

■ 要求確認

要求分析

あき

- 顧客や利用者の要求を明らかにし、仕様化する作業
 - 正確かつ厳密に定義することが重要
 - 設計工程や実装工程において誤ったソフトウェアを作成
 - 設計作業や実装作業のやり直しが発生
 - 要求定義 (Requirements definition)とも呼ぶ
- 要求 (requirements)
 - ソフトウェアを利用して実現したい内容
 - 利用する側から見たソフトウェアが実現すべき目標
 - 機能要求： 開発するシステムが何をするか
 - 非機能要求： 機能要求以外の要求。機能の実現に関する制約を表す (性能、使いやすさ、安全性、保守性、可搬性など)
- 仕様 (specification)
 - 要求を実現するために必要な機能や性能を提供する際の条件や制約
 - 要求を満たすためにソフトウェアが実現しなければならない要件

要求分析の関係者

■ 顧客・利用者

- 要求の発生元
- システム利用者と契約者が異なる場合もある

■ 分析者(システムアナリスト) analyst

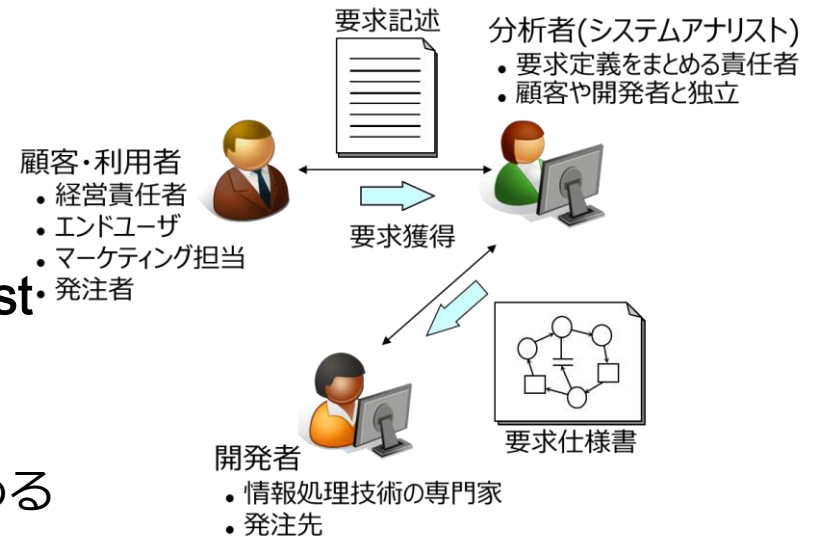
- 要求分析を主導する人
- 利用者と開発者のコミュニケーションギャップを埋める
- 利害関係を調整

■ 開発者

- システムを開発する人
- 技術的観点から非現実的な要求を削除

■ 分析者を置かない場合もある

- 開発者が要求獲得・仕様化も行う



要求分析の作業

- **要求獲得** (Requirements elicitation)
 - 利用者から要求を引き出す
 - 要求定義の初めに行われる
 - 「要求記述」を作成
- **要求記述 (要求仕様化)** (Requirements description/specification)
 - 「要求仕様」を実際に作成する
- **要求確認** (Requirements validation/verification/evaluation)
 - 作成された要求仕様の正しさを検査
 - 要求仕様のレビュー(review)

要求分析の課題

はんえい

■ 利用者の要求を正確かつ厳密に仕様に反映することは困難

1. 顧客や利用者の要求の曖昧さ

- 真の利用者を特定することが困難
- 現状の業務形態やその問題に対する認識が不十分

2. 顧客や利用者の要求の変化

- 初めから要求を完全に把握することは無理
- 利用者自身も要求を明確に認識していない可能性
- 際限ない要求 无限的要求
- 市場や社会情勢、業務環境の変化

3. 利用者と開発者のコミュニケーションギャップ gap

- 背景、知識、言葉の問題
- 利用者は業務の専門家、開発者はシステム開発の専門家
 - 業務知識 = ドメイン知識 (domain knowledge)
- 同じ仕様に対する解釈の違い

確認問題

- ソフトウェアを利用して実現したい内容のことを(1)と呼ぶ。
- (1)には大きく分けて、
システムが何をするかを表す(2)と
システムの機能の実現に関する制約(性能等)を表す(3)がある。
- 要求分析の作業は大きく以下の3つに分けることができる。
 - (4)：利用者から要求を引き出す
 - (5)：要求仕様を実際に作成する
 - (6)：作成された要求仕様の正しさを検査
- 要求分析が難しい理由として、以下のようなものが挙げられる。
 - 顧客の要求が(7)であること。
 - 顧客の要求が(8)すること。
 - 顧客と開発者の間の(9)。

単に用語を覚えるのではなく、要求分析がどのような作業なのか、
なぜ必要なのかを確認してください。

確認問題

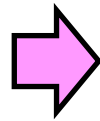
- ソフトウェアを利用して実現したい内容のことを(1)と呼ぶ。
要求
- (1)には大きく分けて、
システムが何をするかを表す(2)と 機能要求
システムの機能の実現に関する制約(性能等)を表す(3)がある。
- 要求分析の作業は大きく以下の3つに分けることができる。非機能要求
 - (4)：利用者から要求を引き出す 要求獲得
 - (5)：要求仕様を実際に作成する 要求記述
 - (6)：作成された要求仕様の正しさを検査 要求確認
- 要求分析が難しい理由として、以下のようなものが挙げられる。
 - 顧客の要求が(7)であること。曖昧
 - 顧客の要求が(8)すること。変化
 - 顧客と開発者の間の(9)。コミュニケーションギャップ

単に用語を覚えるのではなく、要求分析がどのような作業なのか、
なぜ必要なのかを確認してください。

講義内容

■ 要求分析

- 概説



- 要求獲得

- 要求仕様化

- 要求確認

要求獲得

ちゅうしゅつ

1. 要求の抽出

- システムの利用者が何を求めているかを明らかにする
- 複雑な問題を正確に認識することを目指す
- 利用者自身が明文化できない要求も文書化

2. 要求の取舍選択

- すべての要求に応えると経費が膨大に
- オプション^{option}の過不足の防止
- **トレードオフ**の発見と解決 权衡
 - **トレードオフ(trade-off)**：一方を追求すると他方を犠牲にせざるを得ないという関係・状態。

確認問題

■ 要求分析に関する説明として、以下の各文は正しいか。○か×で答えよ。

- 要求抽出において、要求とは、利用者が明示的に述べたものがすべてである。
- 要求抽出においては、分析者と利用者のコミュニケーションが重要である。
- 利用者が出した要求はすべて実現しなければならない。

確認問題

■ 要求分析に関する説明として、以下の各文は正しいか。○か×で答えよ。

- 要求抽出において、要求とは、利用者が明示的に述べたものがすべてである。×
 - 要求抽出においては、分析者と利用者のコミュニケーションが重要である。○
 - 利用者が出した要求はすべて実現しなければならない。×
- 1.利用者自身も要求のすべてを確認しているわけではない
要求は曖昧なこともある →明確化が必要
- 3.コストや技術的に可能か等を考慮しなければならない

要求抽出の技法

■ 資料収集	業務やシステムに関する資料の収集
■ インタビュー	顧客や利用者への質疑
■ アンケート	質問事項に対する回答
■ ブレーンストーミング	自由な意見交換と意見整理
■ 現場観察	業務の体験や観察
■ プロトタイピング	試作品の提供
■ シナリオ scenario	状況ごとの利用例 按照情况的使用例
■ ユースケース	典型的な利用事例 典型的使用例
■ ゴール指向分析	ゴール(目標)の明確化と分割 ^{ぶんかつ}
■ 問題フレーム (problem frame)	機械、世界(問題領域)、要求という 捉え方により問題をパターン化

ブレインストーミング (brainstorming)

- あるテーマに対して参加者が奔放に意見を出し合い、他人のアイデアを発展・改善したりして、たくさんのアイデアを集める方法
- ブレインストーミングの約束事
 - 他人の意見を批判しない
 - 自由奔放な発言を歓迎する
 - 質より量を求める
 - 他人のアイデアを発展・改善させる
- 集まったアイデアの整理にKJ法などが使える
 - アイデア1つをカード1枚に記述
 - 似たアイデアのカードをまとめ、グループを作る
 - グループ間の関連やグループの性質を分析する

シナリオ scenario

- ある状況に限定したシステムの具体的な利用例
- 利点
 - 利用時の様子を記述するものであるため、ソフトウェア開発の専門家でなくとも理解可能
 - 設計理由の文書化・再利用
 - 機能がどのような文脈で使われるのかが分かる
 - 後工程の評価の際に再利用可能
 - 要求分析時に行われる具体例分析において自然に利用できる
 - ワークフロー、業務プロセスをモデル化できる workflow
 - 未知の作業を明確化できる

シナリオ

自然言語によるシナリオの例

タケシはWindows98のノートパソコンを使い始めて約1年のユーザである。彼は、ほぼ毎日パソコンを、主に友達と電子メールをやり取りするために使っている。今日も電子メールを3通書いて送った後、パソコンの電源を切ろうと思った。タケシは、まず[スタート]ボタンをクリックした。表示されたスタートメニューでは、選択項目の下2つ[takeshiのログオフ]と[Windowsの終了]と他の選択項目との間には、線が引かれて区別されていた。ときどき間違えて選択してしまうことがあるので、2つのうち、彼は慎重に[Windowsの終了]を選択した。[Windowsの終了]ダイアログボックスが画面中央に表示された。タケシは「次の方法で終了してもよろしいですか？」という質問を一瞬見た後、4つの選択肢をさっと見た。そのうち、2番目の「電源を切れる状態にする」のボタンがいつもどおり選択されているのを確認し、[はい]をクリックした。小さな効果音とともに画面が暗くなり、その後、自動的にパソコンの電源が切れた。

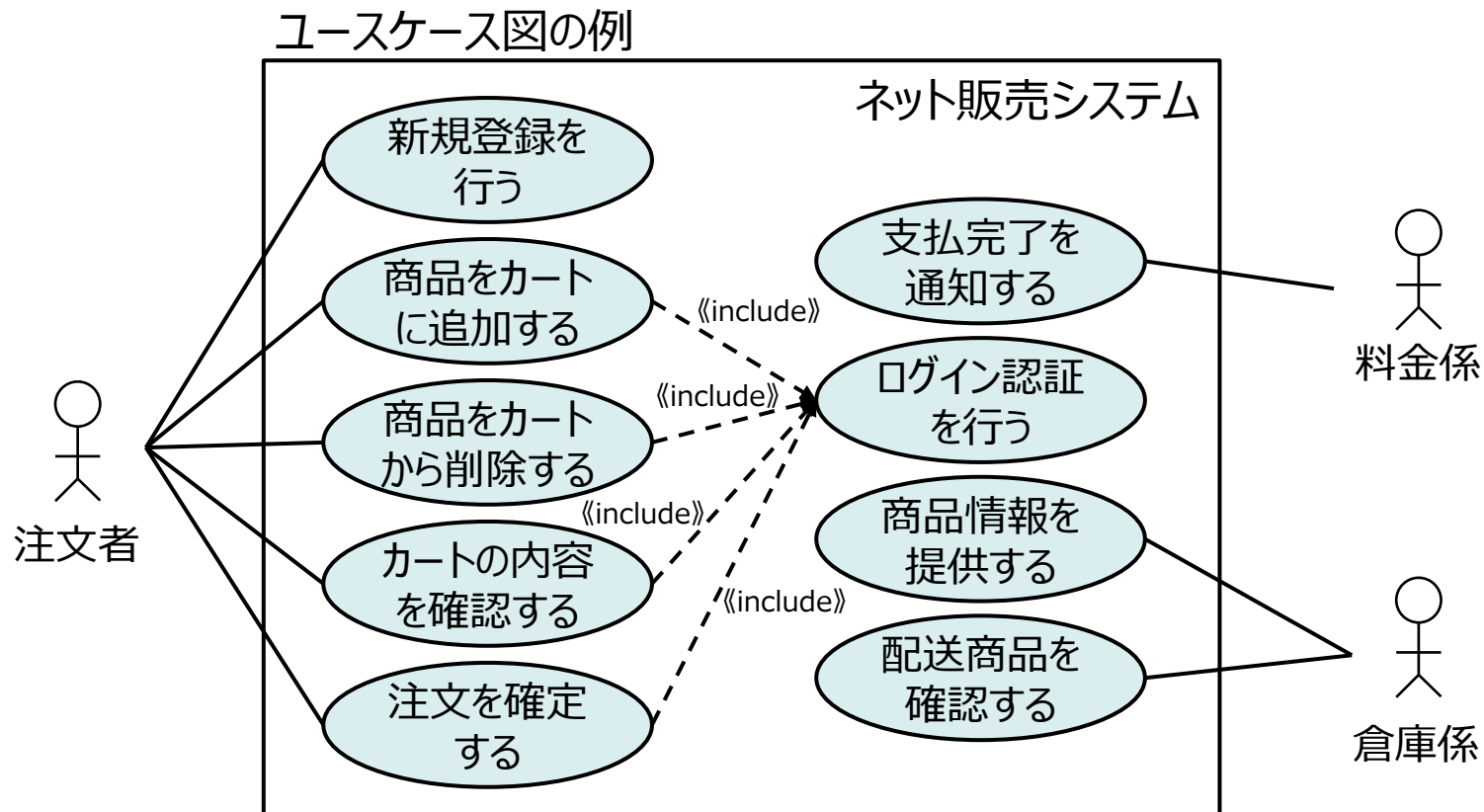
大西淳、郷健太郎「要求工学」共立出版 (2002)

- シナリオの記述方法：
自然言語、絵、動画、フローチャート、代数記述 etc.
flowchart

ユースケース use case

■ 利用者がどのようにシステムを使用するのかを表す
典型的な事例

■ シナリオを一般化したもの



ゴール指向分析

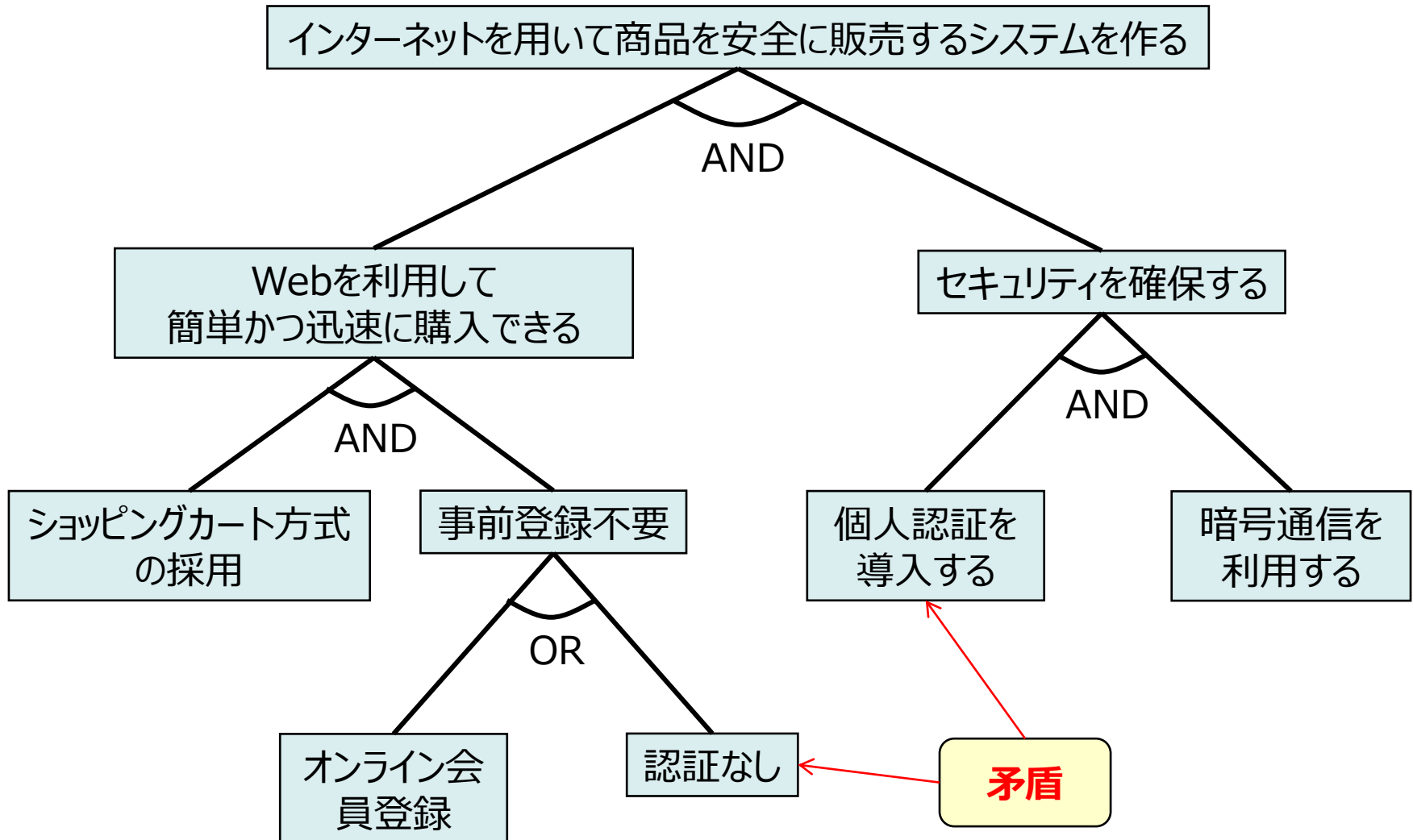
■ 手順

- 達成すべき大きなゴール(目標)を**明確**にする
- ゴールを部分ゴールに**分割**し、ゴール木を作成
 - AND分割：すべてを満たさなければならない
 - OR分割：いずれかを満たさなければならない

■ 利点

- **ゴールや要求を系統的に整理できる**

ゴール木の例



確認問題

- 以下の各文は要求抽出に関連する技法の説明である。それぞれの技法を説明したものか答えよ。
- (1) あるテーマに対して参加者が奔放に意見を出し合い、他人のアイデアを発展・改善したりして、たくさんのアイデアを集める方法。
 - (2) アイデアをカードに記述し、それをグループ化し、グループ間の関連やグループ間の性質を分析する方法。(1)で集まったアイデアの整理に使える。
 - (3) ある状況に限定したシステムの具体的な利用例。
 - (4) 利用者がどのようにシステムを使用するのかを表す典型的な事例。UMLの図的表現としても採用されている。
 - (5) 達成すべき大きな目標を定め、それを分割していき木構造を作る。系統的に要求を整理できる。

確認問題

- 以下の各文は要求抽出に関連する技法の説明である。それぞれの技法を説明したものか答えよ。
- (1) あるテーマに対して参加者が奔放に意見を出し合い、他人のアイデアを発展・改善したりして、たくさんのアイデアを集める方法。**ブレインストーミング**
 - (2) アイデアをカードに記述し、それをグループ化し、グループ間の関連やグループ間の性質を分析する方法。
(1)で集まったアイデアの整理に使える。**KJ法**
 - (3) ある状況に限定したシステムの具体的な利用例。**シナリオ**
 - (4) 利用者がどのようにシステムを使用するのかを表す典型的な事例。UMLの図的表現としても採用されている。**ユースケース**
 - (5) 達成すべき大きな目標を定め、それを分割していき木構造を作る。系統的に要求を整理できる。**ゴール指向分析**

要求の^{しゅしゃ}取舍選択の技法

- 交渉 (negotiation) 合意形成による要求間の矛盾解消
- トリアージ (triage) 要求の優先順位付け
- AHP法 解決案の階層化と点数付け
- デルファイ法 stakeholder
 - ステークホルダーがそれぞれ独自に意見を出し合い、相互参照を行って再び意見を出し合うという作業を繰り返す
 - 全体の傾向や他人の意見がフィードバックされることで、全体の意見が収束する(ことを目指す)

AHP(Analytic Hierarchy Process)法

- 階層分析法、階層化意思決定法とも呼ぶ
- 1971年、T.L. Saatyが提唱した合意形成手法

- 手順
 - 問題を最上位に、代替案を最下位に、
代替案を選ぶための評価基準を中間に置いた階層構造を定義
 - 階層ごとに、要素2つを選び出し、一段上の目的を達成するためにどちらが重要かを数値化する(一対比較)
 - 一対比較の結果から各要素の重みを算出
 - 各階層ごとの重みを合成して、代替案の重みを求める

確認問題

- 以下の各文は要求の取捨選択に関連する技法の説明である。それぞれどの技法を説明したものか答えよ。
- (1) 限られたリソースをどのように割り振るのか決めるために要求に優先順位を付ける方法。医療用語からの転用
 - (2) 問題、代替案、評価基準の階層構造を定義し、階層ごとに重要度を一対比較し、その結果を合成していくことでどの案を採択するか決定する
 - (3) ステークホルダーがそれぞれ独自に意見を出し合い、相互参照を行って再び意見を出し合うという作業を繰り返し、全体の意見を収束させていく

確認問題

- 以下の各文は要求の取捨選択に関連する技法の説明である。それぞれの技法を説明したものか答えよ。
- (1) 限られたリソースをどのように割り振るのか決めるために要求に優先順位を付ける方法。医療用語からの転用 **トリアージ**
 - (2) 問題、代替案、評価基準の階層構造を定義し、階層ごとに重要度を一対比較し、その結果を合成していくことでどの案を採択するか決定する **AHP**
 - (3) ステークホルダーがそれぞれ独自に意見を出し合い、相互参照を行って再び意見を出し合うという作業を繰り返し、全体の意見を収束させていく **デルファイ法**

講義内容

■ 要求分析

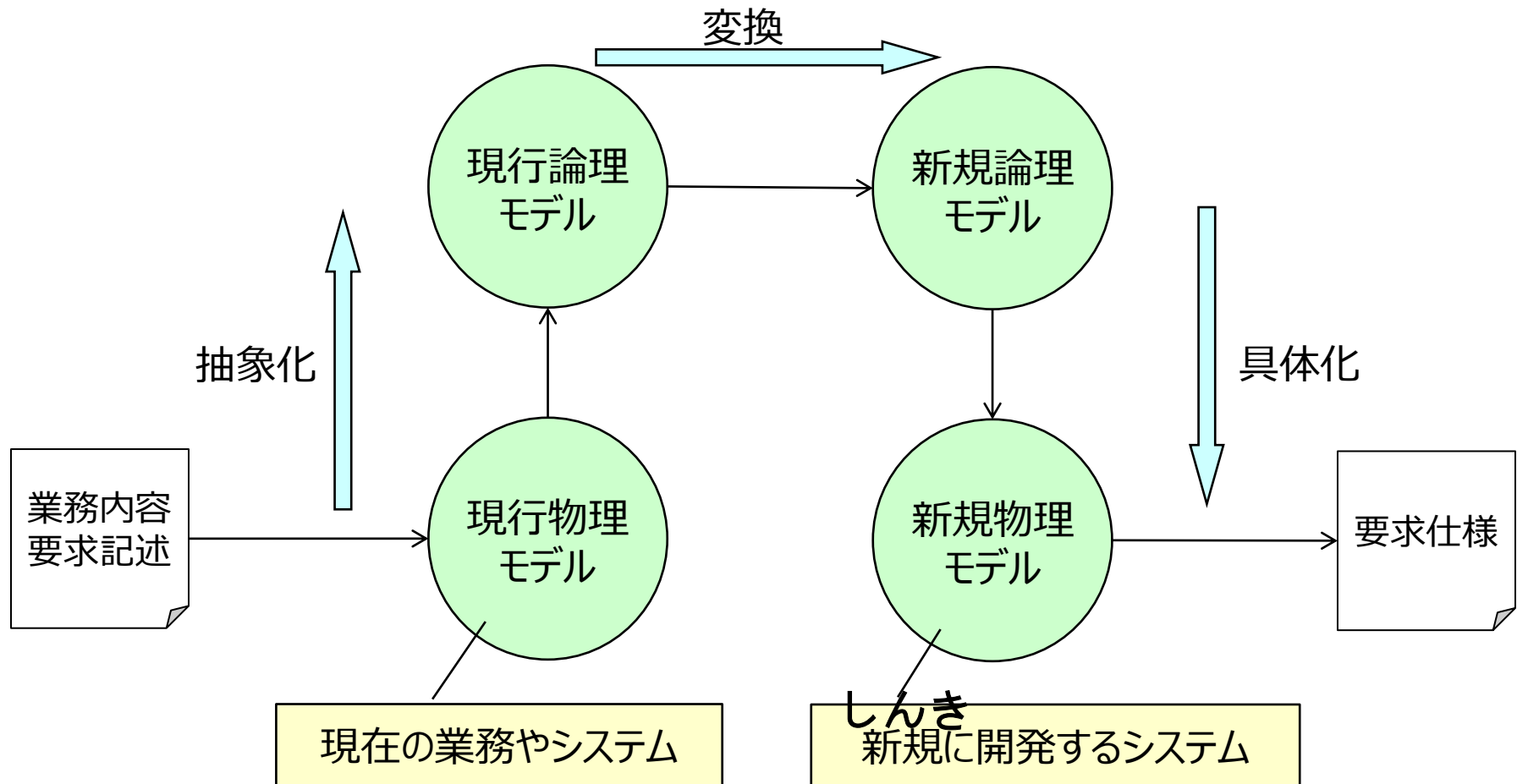
- 概説

- 要求獲得

- ➡ ■ 要求仕様化

- 要求確認

構造化分析における要求仕様化手順



論理的：どのような情報が必要であるかという要件（本質的）

物理的：その情報を得るための仕組みに対する要件（具体的）

構造化分析における要求仕様化手順

1. 現行物理モデルの構築

現状の業務や現行システムの要求記述を分析

- ありのまま(as-is)に具体的に表現

2. 現行論理モデルの構築

本質的な機能や問題の洗い出し

- 本質的でない部分（慣例的な処理や特定の事情による処理）を除外
 - 人名、媒体、日時、金額、数量など

3. 新規論理モデルの構築

新たに開発するシステムで解決すべき問題の洗い出し

- 追加あるいは削除する機能の特定
- あるべき姿(to-be)を表現

4. 新規物理モデルの構築

システムが満たすべき制約や性能を^{ふよ}付与

5. 要求仕様の記述

要求仕様記述言語

- 自然言語(natural language)
 - e.g., 日本語、中国語、英語
- 形式的言語(formal language)
 - e.g., Z, VDM, CafeOBJ
- 制限言語(controlled language)
 - e.g., 擬似^{きじ}コード、X-JRDL
- 図式言語(graphic language)
 - e.g., UML、DFD

自然言語による要求の記述

■ 人間が日常使う言葉を使用

■ 利点

- 日常用いている言語であるため、特別な学習なしに書いたり読んだりできる
- 分野によらず利用可能

■ 欠点

■ 曖昧さ(定性的な表現等)

- e.g., システムの応答性が高いこと

■ 語句の省略や代名詞、多義語の使用

- e.g., ATM → Automatic Teller Machine (現金自動預け払い機)?
Air Traffic Management (航空交通管制)?

形式的言語による要求の記述

■ 文法や意味が厳密に定義された人工言語を使用

■ 利点

■ 曖昧さがない

■ 仕様の正しさを数学的に証明できる

■ 計算機処理が可能
→ 作成支援・理解支援等

```
object STACK-1 is
  sort Stack
  op create : -> Stack .
  op push : Element, Stack -> Stack .
  op pop : Stack -> Stack .
  op top : Stack -> Element .
  op is_empty : Stack -> Boolean .
  var E : Element .
  var S : Stack .
  eq pop (push (E,S) ) = S .
  eq top (push (E,S) ) = E .
  eq is_empty (create) = true .
endo
```

形式的言語によるスタックの仕様例

■ 欠点

■ 学習コスト(習得しなければ読み書きできない)

■ 非専門家には使いにくい

■ 特定の問題領域のみを対象としている

制限言語による要求の記述

■ 自然言語と形式言語の中間

- 自然言語に制約を加える

■ 利点

- 比較的理解しやすく、書きやすい
- 自然言語より計算機処理が容易

■ 欠点

- 若干の習得が必要
- 文法や語彙に制限がある
- 問題領域に制約がある

```
do ファイルを1行読み込む
  while ファイルの終わりでない
    if 得点<60
      then 該当行を出力する
    fi
  od
```

制限言語による仕様例

図式言語による要求の記述

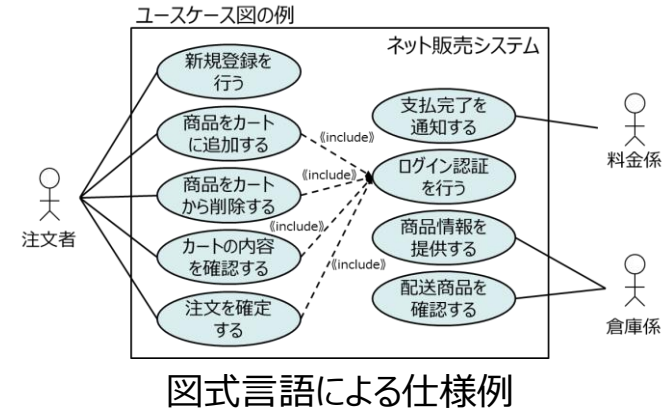
■ 図を用いて仕様を表現

■ 利点

- 直感的にわかりやすい
 - 記法が定められているため、支援ツールも比較的作りやすい
- ひかく

■ 欠点

- 表現対象が限定される
- 表現したい内容が表現できないこともある
- 正確に理解されないこともある



確認問題

- 以下の各文は、要求仕様の記述における自然言語、形式的言語、図式言語の使用について述べたものである。それぞれ、どの言語について述べたものか答えよ(答えは一つとは限らない)
 - 普段人間が使用している言語であるため、特別な学習をせずとも読み書きできる。
 - 語句の省略等により曖昧さが混入しやすい。
 - 正しさを数学的に証明できる。
 - 図を用いるため、直感的に理解しやすい。
 - 記法が定められているため、支援ツールが作りやすい。
 - 表現対象が限定される。

確認問題

- 以下の各文は、要求仕様の記述における自然言語、形式的言語、図式言語の使用について述べたものである。それぞれ、どの言語について述べたものか答えよ(答えは一つとは限らない)
 - 普段人間が使用している言語であるため、特別な学習をせずとも読み書きできる。**自然言語**
 - 語句の省略等により曖昧さが混入しやすい。**自然言語**
 - 正しさを数学的に証明できる。**形式的言語**
 - 図を用いるため、直感的に理解しやすい。**図式言語**
 - 記法が定められているため、支援ツールが作りやすい。
 - 表現対象が限定される。**形式的言語 & 図式言語**
形式的言語 & 図式言語

講義内容

■ 要求分析

- 概説

- 要求獲得

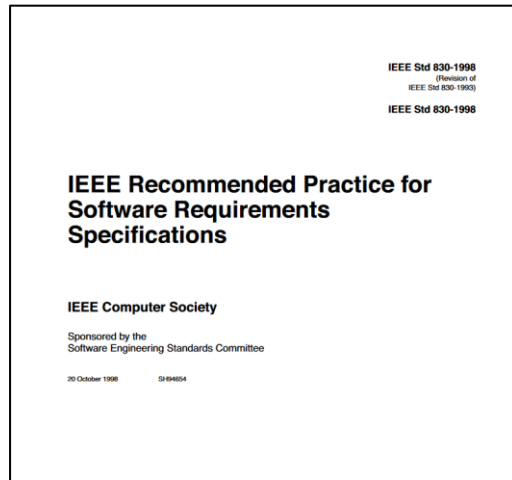
- 要求仕様化

- ➡ ■ 要求確認

要求確認

■ 要求に関する品質特性(IEEE Std. 830-1998)

- 妥当性（正当性）： 実現すべき要求を含んでいるか
- 非曖昧性： すべての要求が一意に解釈できるか
- 完全性： 必要な情報がすべて含まれているか
- 無矛盾性（一貫性）： 要求が互いに矛盾しないか
- 重要度と安定性の順位付け： どの要求を優先的に扱うか
- 検証可能性： 完成したシステムに対して検証可能か
- 変更可能性： 要求の一部だけを変更可能か
- 追跡可能性： 要求仕様とその他の成果物との対応が取れているか
 - 後方追跡可能性：
各要求について、背景、理由、意図が容易に参照可能
 - 前方追跡可能性：
要求仕様に基づいて作成された
設計文書、ソースコード、マニュアルが容易に参照可能





参考文献

- 「ソフトウェア工学」
高橋直久、丸山勝久 著、森北出版、2010
- 「要求工学」
大西淳、郷 健太郎 著、共立出版、2002
- 「要求分析とプロジェクト管理 (part1)」
手塚太郎、<http://xi.kc.tsukuba.ac.jp/lec/RAPM1.pdf>
- 「発想法」
川喜田二郎 著、中公新書、1967