

# 既存システムの ユースケース図を利用した ステークホルダと 非機能要求の獲得法

2004年8月3日

海谷 治彦 (信州大学)

# 目次

- 研究の背景と目的
- 獲得法の入出力とその利用法
- 仮定と基本的なアイデア
  - ゴール指向分析
  - GQM
- 獲得法の概要
- 例題
- まとめと今後の課題

# 背景

- システム開発を行う場合，既存の類似システムを参考に開発を始める.
  - 同業他社のシステムも含めて参考にする.
- しかし，既存システムとの違いは必要.
  - ビジネス上の優位を得るため.
  - 著作権の問題.
  - システムが利用される文脈が既存のものと若干違う.
- 上記は要求仕様書を書き始める前に，明確化しておかなければいけない事項である.

# 目的

既存システムの考え方を参考にしつつ、

- 既存システムとの差別化点(新システムのゴール)を明確にし、
- 新システムの利用文脈を明確にする。

既存システム

⇒ ユースケース図(UCD)による要求仕様書

Use Case Diagram

差別化点

⇒ 非機能要求 (NFR; 主に性能や品質の要求)

Non-Functional Requirements

利用文脈

⇒ ステークホルダ (利害関係者)

Stakeholders

# 獲得法の入出力

- 入力
  - 新システムを欲する顧客
  - そのシステムの種類(ドメイン)
  - 類似システムのUCDを可能な限り複数個
- 出力
  - 業務の目標 (他との差別化項目)
    - 非機能要求 (NFR)
      - これは主に機能要求(ユースケース UC)の修飾情報となる.
  - システムの利害関係者(ステークホルダ)
    - 敵対関係のモノを含む.

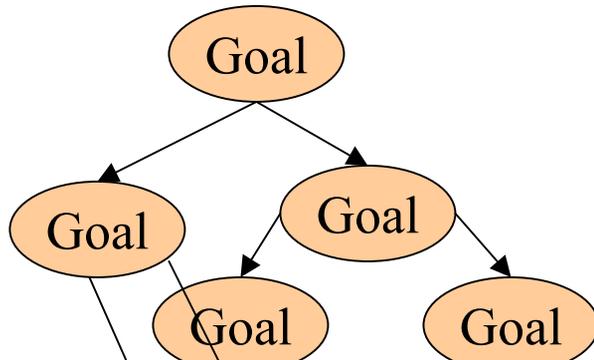
# 出力の利用法

- 要求仕様書の記述に使う。
  - 作成するシステムの利点・特徴が明確になっている。
    - 機能毎にこういった非機能的な特徴を「売り」にし、何故、そのような売りが必要なのかを述べられる。
  - 情報獲得先が特定されている。
    - ステークホルダとなる人物からの情報収集をもとに要求を固められる。
- 出力ではないが、参考にした既存UCDをたたき台に要求仕様書を記述できる。

# 仮定とアイデア

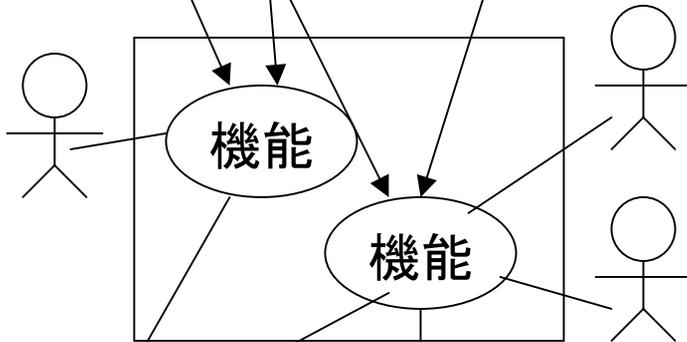
- システム(もしくはビジネス)のゴールからトップダウンにシステムの差別化特徴を得るのは困難.
  - むしろ, 具体的な差異からゴールをボトムアップに決める.
- 人は変化に対しては敏感.
  - 要求項目が変化することで
    - 自分(や他人)がステークホルダだったことに気づく.
    - ある特徴がシステムの差別化につながることに気づく.
- 類似システム(whatもしくはhow)の比較から差別化につながる特徴と, その変化(の方向)が識別しやすいだろう.
- その特徴を一般化することでビジネス上のゴールを認識することができるだろう.

# 仮定の具体化: ゴール指向分析



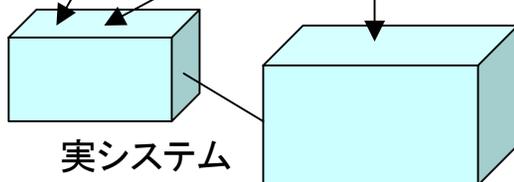
何故, そのシステムがいるか(why)  
NFR等はココでは直接に記述される

Step 2



システムは何をするか(what)  
ユースケース図で表現  
NFRも何からの機能やパラメ  
タに展開されている.

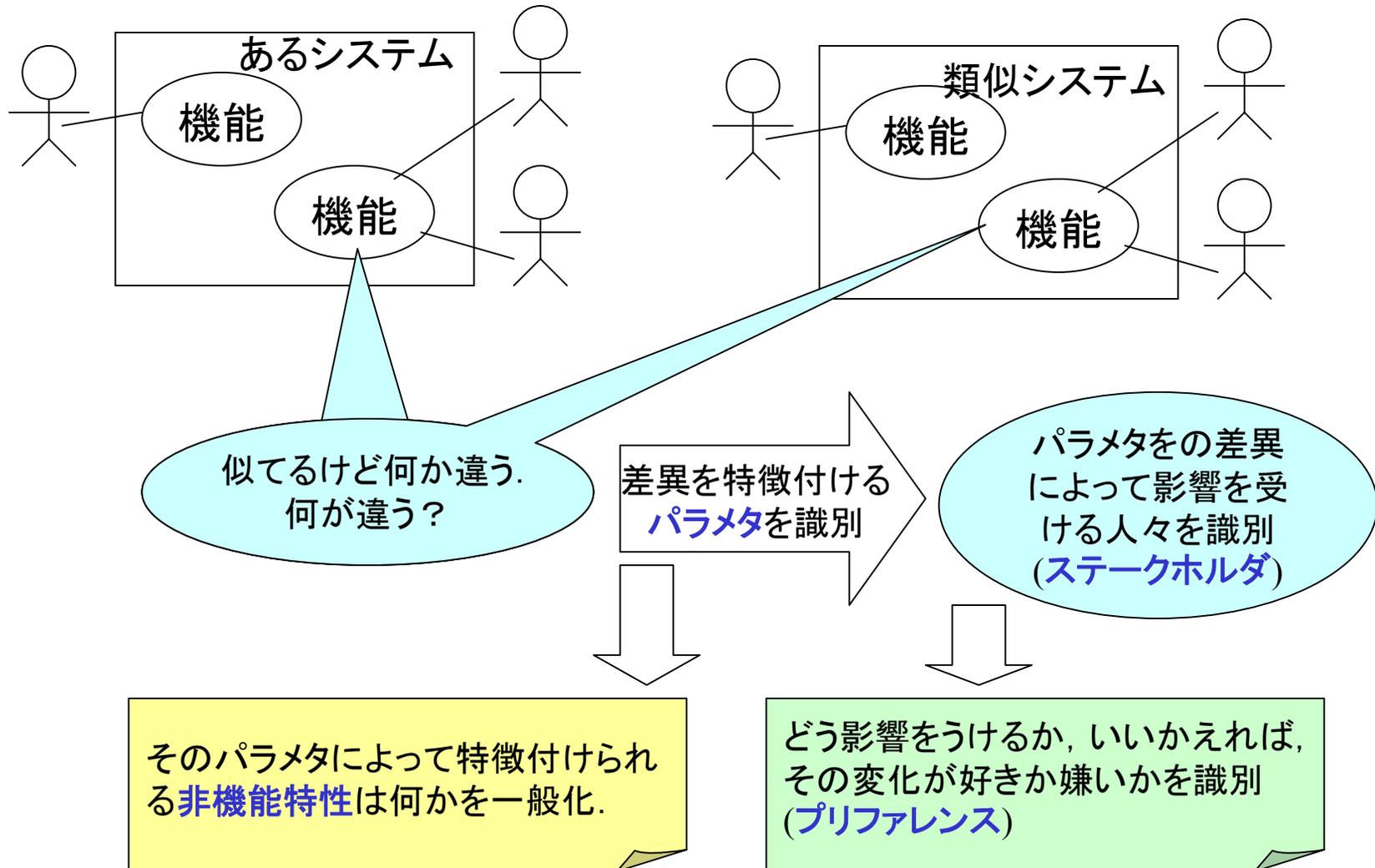
Step 1



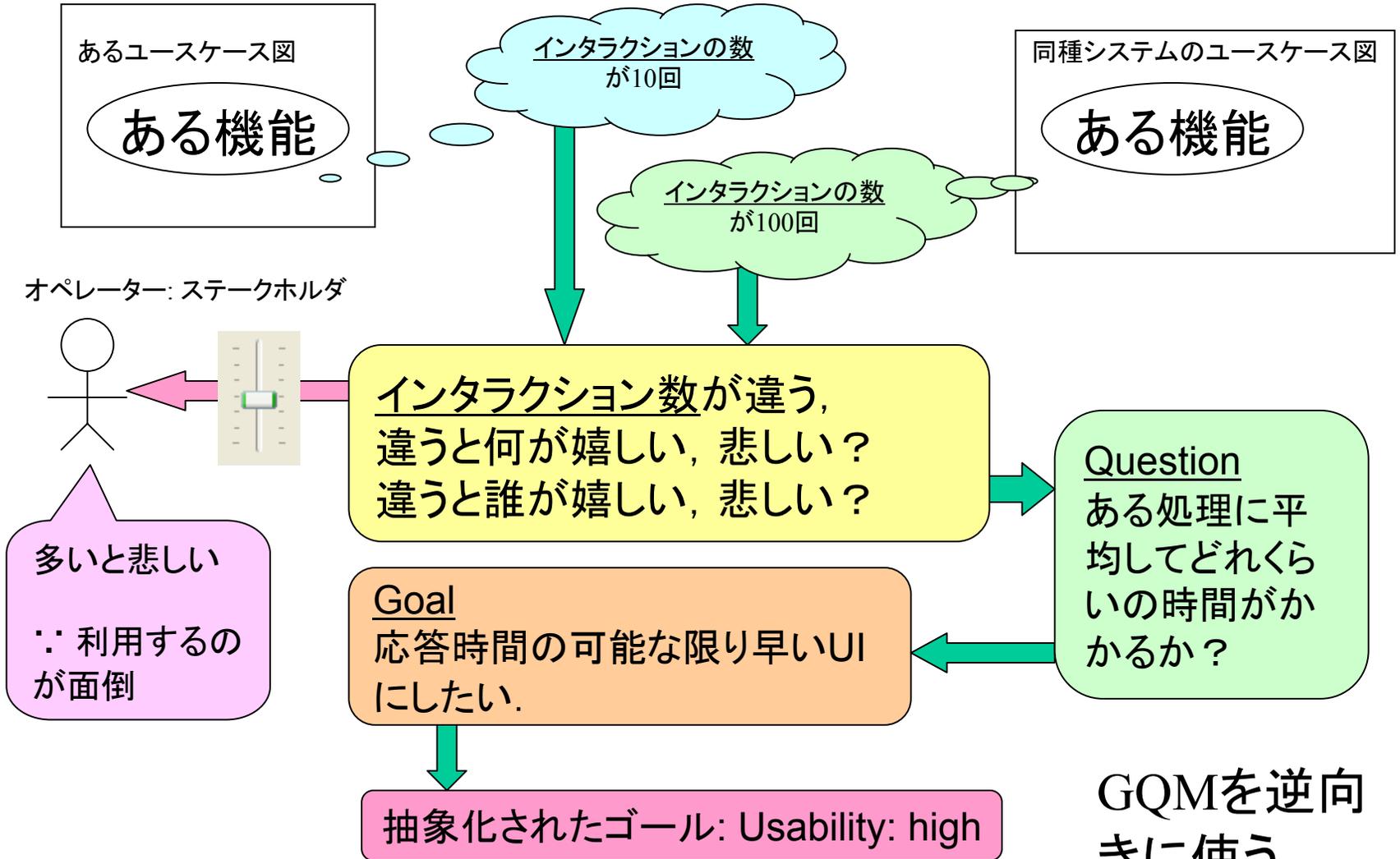
どのように実現するか(how)  
本研究では範囲外

Step 3

# ゴールとステークホルダ発見の指針



# 例

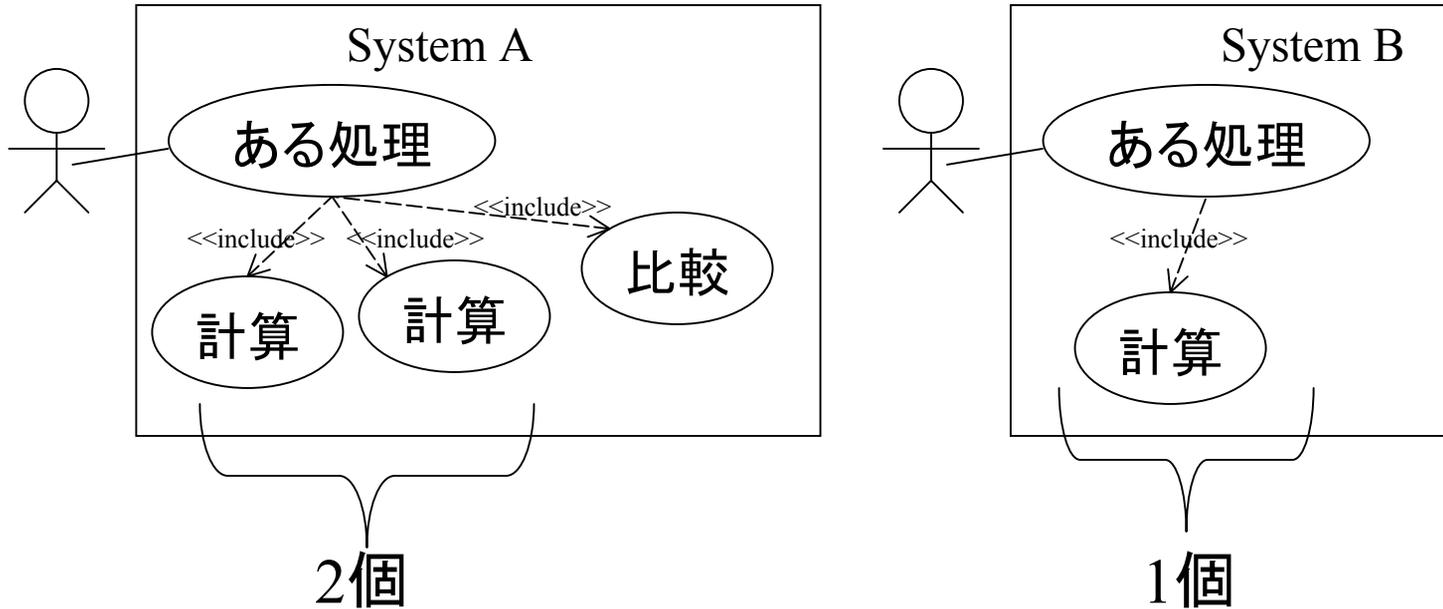


GQMを逆向きに使う

# 「同種」ユースケースの識別

- 同種システムのユースケース図は事前に収集しておく必要がある。
- 同種のシステムについては直感的に集めるしかない。(例: メールクライアント等)
  - 同種のアクター, ユースケース(UC)を含む物とか。
- 同種のアクター, ユースケース
  - 語彙をもとに主観的に判断するしかない。
- ユースケース周囲
  - そのUCに直接繋がる他のUCやアクター
    - UCについては include, extend のどちらかで関連付け。
  - コレの差異によってユースケースの差異を識別する。

# 例



Variableとして「計算の回数」

Typeとして「nat」

Preferenceとして、「たくさん」もしくは「数回」

NFRとして、「信頼性」

Stakeholder「顧客」のPreferenceは「うれしい」∵ 正確な結果が得られる

Stakeholder「開発者」のPreferenceは「悲しい」∵ めんどい

# 獲得法の入出力

- 入力
  - 新システムを欲する顧客
  - そのシステムの種類(ドメイン)
  - 類似システムのUCDを可能な限り複数個
- 出力
  - 業務の目標 (他との差別化項目)
    - 非機能要求 (NFR)  
これは主に機能要求(ユースケース UC)の修飾情報となる.
  - システムの利害関係者(ステークホルダ)
    - 敵対関係のモノを含む.

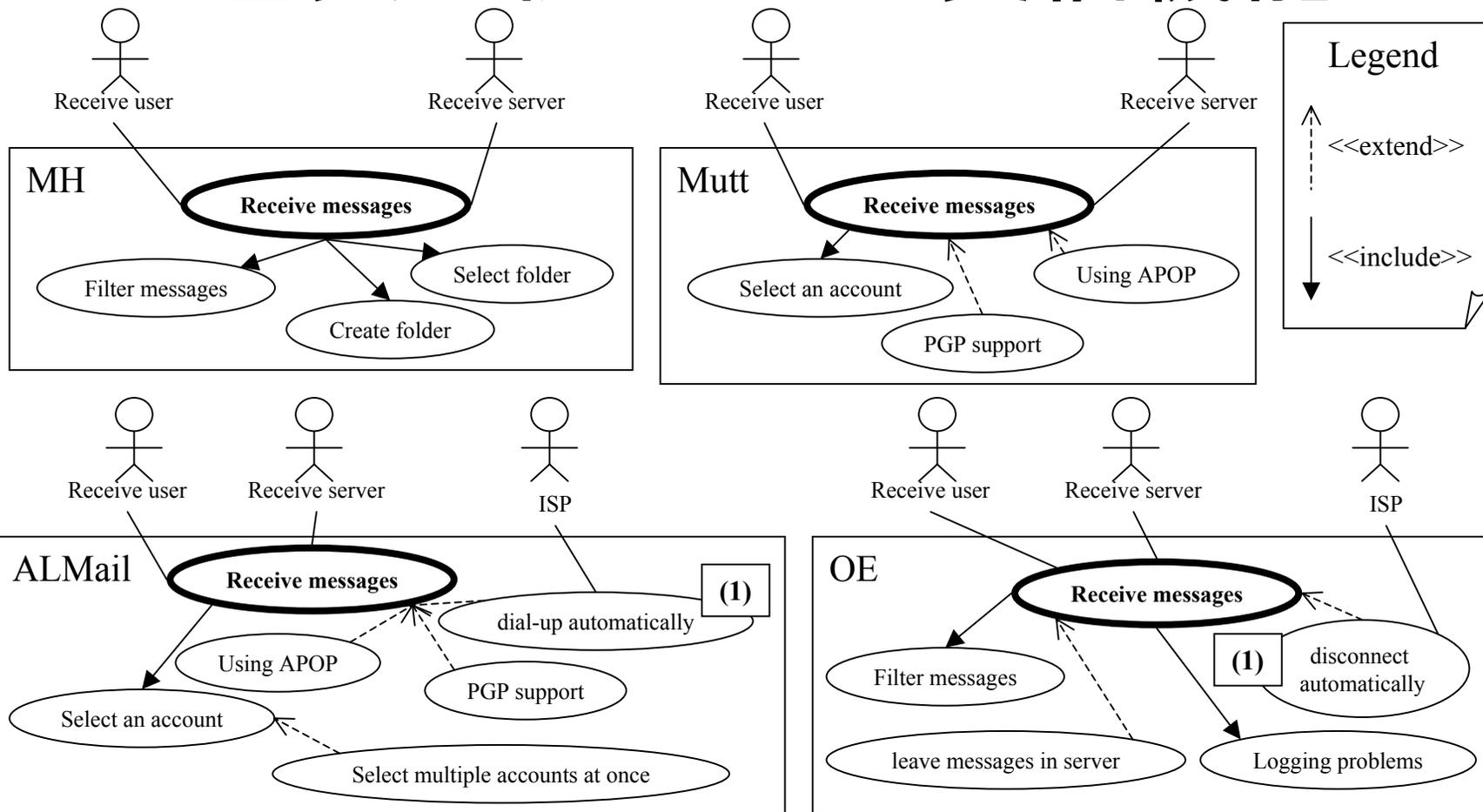
# 獲得手順の概要

1. 類似分野のシステムのUCDを複数集める.
  - UCDが無ければ, マニュアルを見て記述する.
2. UCDから類似UCを識別.
3. 周囲の情報に基づき類似UCの差異を識別.
4. 差異を特徴付ける変数と型を識別.
5. 変数の変化に影響を受けるステークホルダとそのプリファレンス(好き嫌い)を識別.
6. NFRを識別.
  - ある変数がある値に変化することが好ましいことを, 抽象化する.
  - 抽象化には既存のNFRの分類を参考にする.  
ISO/IEC9126やNFR framework[8]を利用.

# 例題

- ドメイン: メールエージェント (Mail User Agent)
- 既存のメールクライアント
  - 1. Outlook Express: Windowsで標準
  - 2. MH: コマンドベースの古いメールクライアント, UNIX系
    - 私は主にこれを使っています.
  - 3. AL-Mail: Windows, 日本固有
  - 4. Mutt: テキストベース, 対話型
    - 他の筆者はこれを使っている.
- 大体機能は似ているが, それぞれに特徴が違う.

# 差異の例: メール受信機能

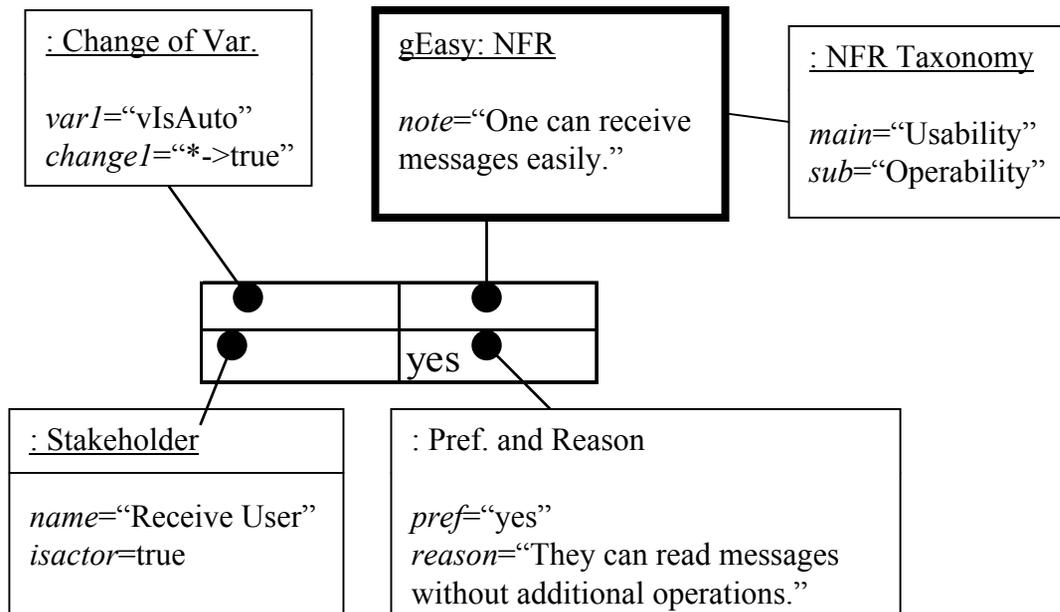


回線接続機能が受信機能に包含されている点が差異

# vIsAutoの変化によるPreference

変数 vIsAuto: boolean

回線を自動的に接続・切断するか否か？



# まとめ

- 要求仕様記述の前に、ステークホルダと(他のシステムとの)差別化項目を明確にするための手法を提案。
  - 要求記述のための材料を準備できる.
- UCDの構造的な差異とNFR分類を手がかりに、主観的な判断を支援.
- メーカーの例に適用した.

# 今後の課題

- 支援ツールの構築
  - 現時点では識別したことの記録・管理ツールではない.
  - 要求記述ツール等の連携
- 識別法に関する積極的な計算機支援
  - オントロジによる語彙に関する支援
  - ユースケース内記述の利用
  - NFRパターンの強化