

## ビジネスイノベーション研究部会 / SOA部会 部会紹介

2009年1月9日(金)

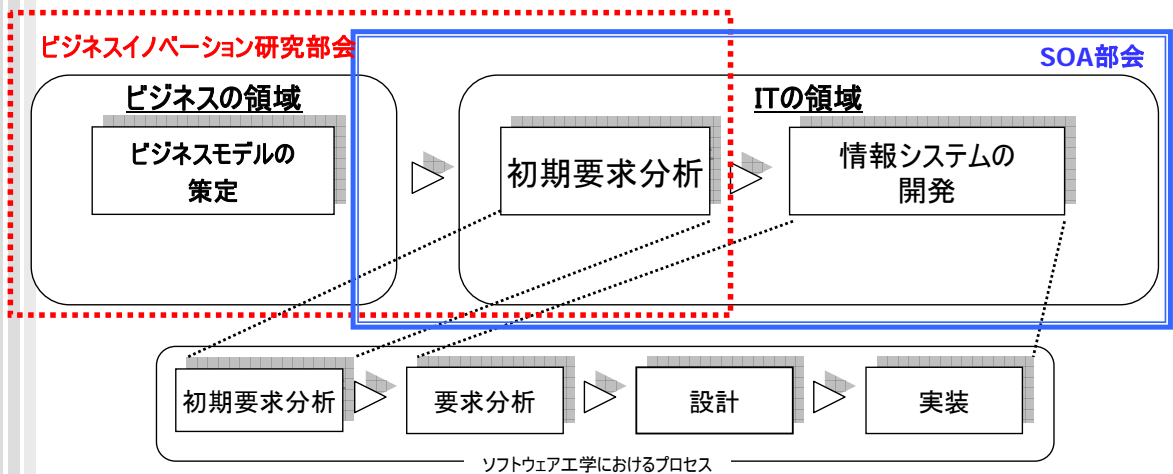
XMLコンソーシアム  
ビジネスイノベーション研究部会 / SOA部会

Copyright © XML Consortium 2008 All rights reserved.

## ビジネスイノベーション研究部会 / SOA部会 紹介

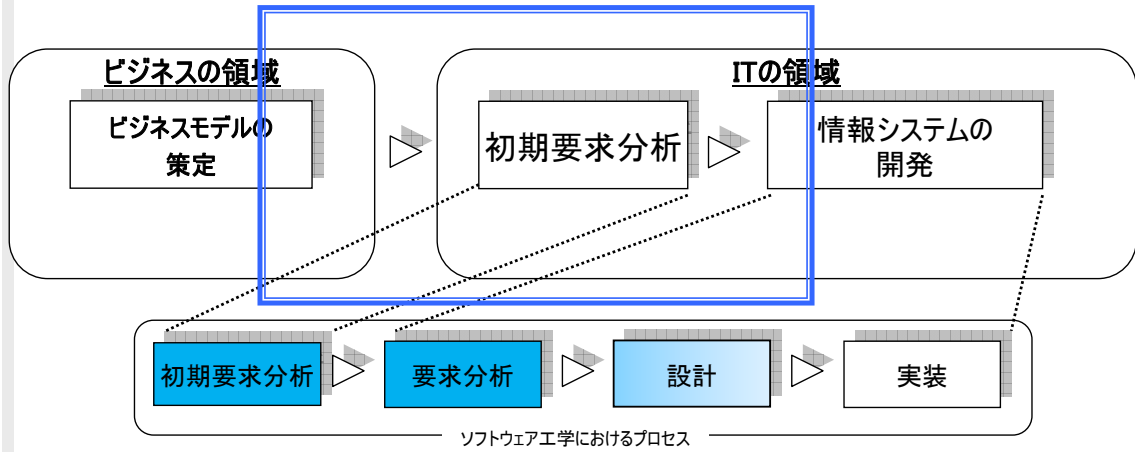


- 両部会では、ビジネス戦略に基づくSOAベースの情報システム開発を調査・研究している。



- **ビジネスイノベーション研究部会 活動内容**
  - ビジネスの目標や計画を反映し、業務とITを関連づける方法
  - エンドユーザ主導の情報活用の形態と必要な技術の研究
- **BI研 WG**
  - ※法によるサービス分析の為のモデリングWG

- **SOA部会 活動内容**
  - 抽象的なSOAの概念から具体的なビジネス面でのメリット、システム設計/実装の方針や方法、Webサービスの有効活用方法等を導き出す
- **SOA部会 WG (2008年度は4つのWG + 部会共通活動)**
  - SOAサービス設計WG, SaaS II WG, SOAクライアント環境, Ws-Transaction技術調査WG
  - 部会共通活動: サービス・モデリング・ハンズオン



(意図的に空白)

## ※法を用いたゴール指向分析とサービス設計 ビジネスプロセス分析の補完への期待

2009年1月9日

XMLコンソーシアム ビジネス・イノベーション研究部会／SOA部会

住友セメントシステム開発株式会社 加藤昌宏

日本アイ・ビー・エム株式会社 根本和郎

Copyright © XML Consortium 2008 All rights reserved.

## メンバー一覧(五十音順、敬称略)



- 芦田尚人 (株)ブレインワークス
- 新井祐也 日本ユニシス(株)
- 飯沢篤志 (株)リコー
- 稲生章人 NECソフト(株)
- 尾々野正和 日立ソフトウェアエンジニアリング(株)
- 何翁径迪 (株)アイ・ティ・フロンティア
- 加藤昌宏 住友セメントシステム開発(株)
- 小林茂 日本ユニシス(株)
- 坂下秀彦 (株)JIEC
- 柴田昌宏 日本電気(株)
- 瀬田晃彦 日本ユニシス(株)
- 妹尾正美 NTTソフトウェア(株)
- 茶谷謙一 (株)日立製作所
- 根本和郎 日本アイ・ビー・エム(株)
- 橋本好史 (株)インテックシステム研究所
- 日力俊彦 日本IBMシステムズ・エンジニアリング(株)
- 牧野友紀 日本ユニシス(株)
- 町田寛 (株)インテックシステム研究所
- 横田治樹 NECソフト(株)



## i\*法を用いた SOA設計の為の要求分析

Copyright © XML Consortium 2008 All rights reserved.



### 初期要求開発の必要性



- “IT does not matter”  
邦訳「ITにお金を使うのは、もうおやめなさい」
- iTunes, iPod, Wii,  
差別化要因はテクノロジーではなく、  
商品設計、Wantsの詳細発掘



ARTICLE

The easiest way  
to invest in IT today?  
Less may be more.

IT Doesn't Matter  
by Nicholas G. Carr

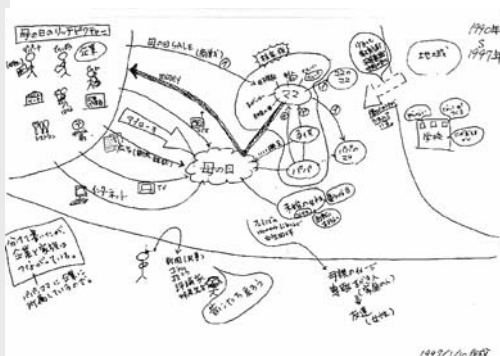
# i\* 法おさらい

- 簡潔な表記法
- 低い学習コスト
  - リッチピクチャーに似た視覚化記法
  - 少なくとも VDM++ や Z よりは直感的
- モデリングの系譜、2流派
  - “UML as a sketch”派 ← ポンチ絵で人が見やすい  
i\*法はこちら
  - “MDA”派 ← Z,OCL,VDMの様な専用言語  
基本はメタプログラミング  
ある種のパターンを書き出す
- 要求の細分化が目的、人が分かること

# 記号化言語 (スケッチ)

表記ルールに縛られずに  
課題を視覚化した図  
低い学習コスト、人にとって理解しやすい

## リッチピクチャー



## マインドマップ



課題を言語化したもの  
 厳格なルール  
 Machine readable

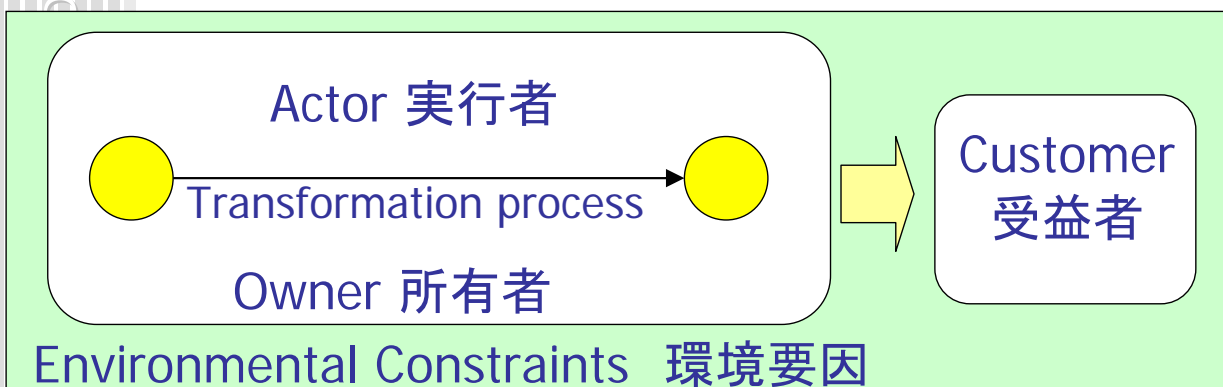
## VDM言語

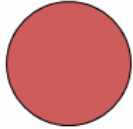




functions  
 Add: Name \* Email \* RegisterBook -> RegisterBook  
 Add(name, email, book) ==  
 book munion {name |-> email}  
 pre name not in set dom(book);  
  
 Delete: Name \* RegisterBook -> RegisterBook  
 Delete(name, book) ==  
 {name} <-: book  
 pre name in set dom(book);  
  
 Number: RegisterBook -> nat  
 Number(book) ==  
 card dom(book)

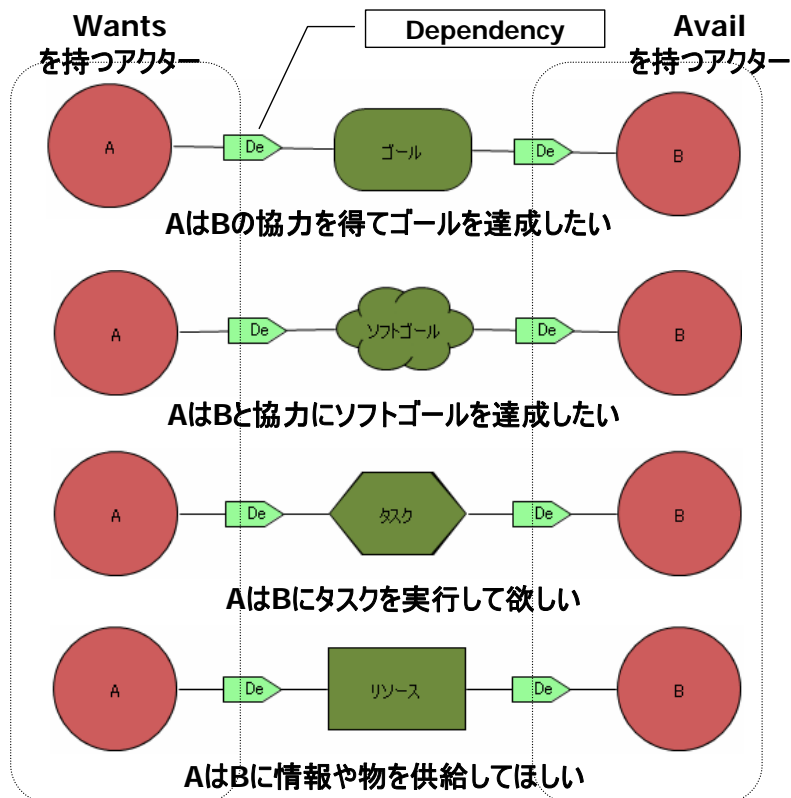
## Z言語

$$\begin{aligned}
 &g_{i1} : \mathbb{P}(e_{i1} \times f_i); \dots; g_{i n_i} : \mathbb{P}(e_{i n_i} \times f_i) \\
 &\vdots \\
 &g_{r1} : \mathbb{P}(e_{r1} \times f_r); \dots; g_{r n_r} : \mathbb{P}(e_{r n_r} \times f_r) \\
 &| (\forall u : e_{i1} \bullet \exists_1 x : g_{i1} \bullet x.1 = u) \wedge \dots \wedge (\forall u : e_{i n_i} \bullet \exists_1 x : g_{i n_i} \bullet x.1 = u) \\
 &:\wedge \\
 &(\forall u : e_{r1} \bullet \exists_1 x : g_{r1} \bullet x.1 = u) \wedge \dots \wedge (\forall u : e_{r n_r} \bullet \exists_1 x : g_{r n_r} \bullet x.1 = u) \\
 &:\wedge \\
 &(\forall u, v : e_{i1} \mid g_{i1} u = g_{i1} v \bullet u = v) \wedge \dots \wedge (\forall u, v : e_{i n_i} \mid g_{i n_i} u = g_{i n_i} v \bullet u = v) \\
 &:\wedge \\
 &(\forall u, v : e_{r1} \mid g_{r1} u = g_{r1} v \bullet u = v) \wedge \dots \wedge (\forall u, v : e_{r n_r} \mid g_{r n_r} u = g_{r n_r} v \bullet u = v) \\
 &\forall b_1, b_2 : \mathbb{N} \bullet \\
 &(\forall w : f_i \mid \\
 & \quad (b_1 = 1 \wedge w = h_{i1} \vee \dots \vee b_1 = m_1 \wedge w = h_{i m_1} \vee \\
 & \quad \quad b_1 = m_1 + 1 \wedge w \in \{x : g_{i1} \bullet x.2\} \vee \dots \vee b_1 = m_1 + n_1 \wedge w \in \{x : g_{i n_i} \bullet x.2\}) \\
 & \quad \wedge (b_2 = 1 \wedge w = h_{i1} \vee \dots \vee b_2 = m_1 \wedge w = h_{i m_1} \vee \\
 & \quad \quad b_2 = m_1 + 1 \wedge w \in \{x : g_{i1} \bullet x.2\} \vee \dots \vee b_2 = m_1 + n_1 \wedge w \in \{x : g_{i n_i} \bullet x.2\}) \bullet \\
 & \quad \quad b_1 = b_2) \wedge \\
 &:\wedge \\
 &(\forall w : f_r \mid \\
 & \quad (b_1 = 1 \wedge w = h_{r1} \vee \dots \vee b_1 = m_r \wedge w = h_{r m_r} \vee \\
 & \quad \quad b_1 = m_r + 1 \wedge w \in \{x : g_{r1} \bullet x.2\} \vee \dots \vee b_1 = m_r + n_r \wedge w \in \{x : g_{r n_r} \bullet x.2\}) \\
 & \quad \wedge (b_2 = 1 \wedge w = h_{r1} \vee \dots \vee b_2 = m_r \wedge w = h_{r m_r} \vee \\
 & \quad \quad b_2 = m_r + 1 \wedge w \in \{x : g_{r1} \bullet x.2\} \vee \dots \vee b_2 = m_r + n_r \wedge w \in \{x : g_{r n_r} \bullet x.2\}) \bullet \\
 & \quad \quad b_1 = b_2)
 \end{aligned}$$

- |                         |             |
|-------------------------|-------------|
| •Customer               | 影響を受ける人     |
| •Actor                  | 実行者         |
| •Transformation process | 何が起きるのか     |
| •World view(世界観)        | そうすることの価値   |
| •Owner                  | プロジェクト・オーナー |
| •Environment constrains | 環境要因        |

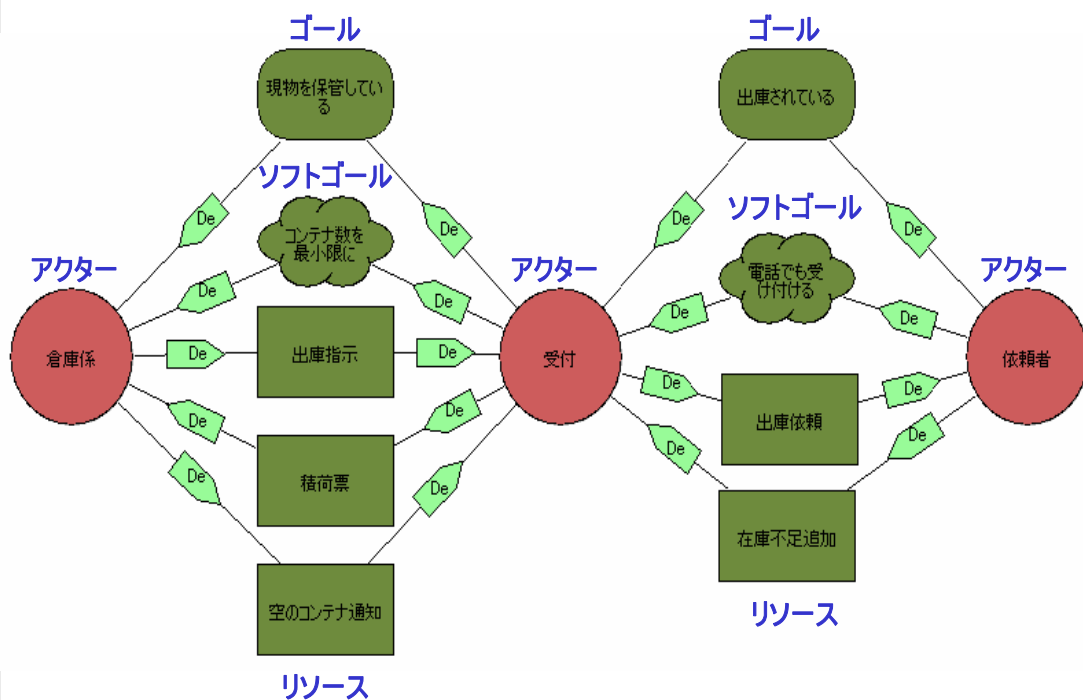


-  **アクター**: ゴールを達成するために行動する実体.
-  **ゴール**: 達成したい事柄の条件や状態.
-  **ソフトゴール**: 達成の可否が明確に判断できない条件や状態.
-  **タスク**: ゴールを達成する特定の手順.
-  **リソース**: ゴール達成(タスク遂行)に利用する物や情報.



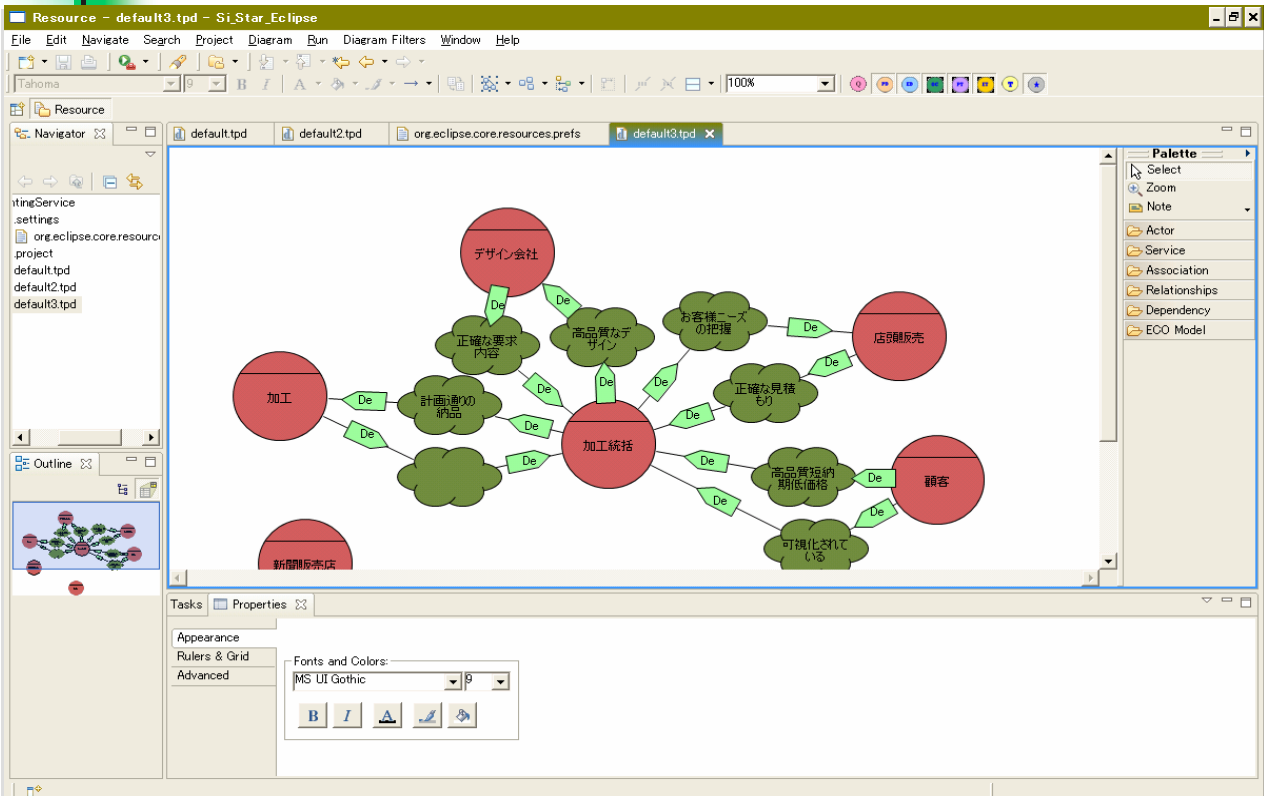
- アクター間の利害関係の分析が可能
  - ゴールは何か (ゴールサービスモデル)
  - 必要なリソースの明確化
  - 何を協力して欲しいのか (タスク分割)
  - どのようにやって欲しいのか (早く、安く、・・・)

## Dependencyモデル例 (酒屋モデル)





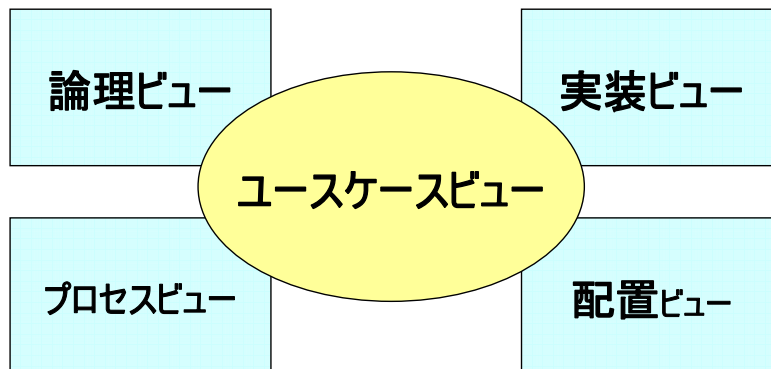
# 作画ツール : Si\_Star\_Eclipse



## 4+1 view (RUP)に於けるユースケース



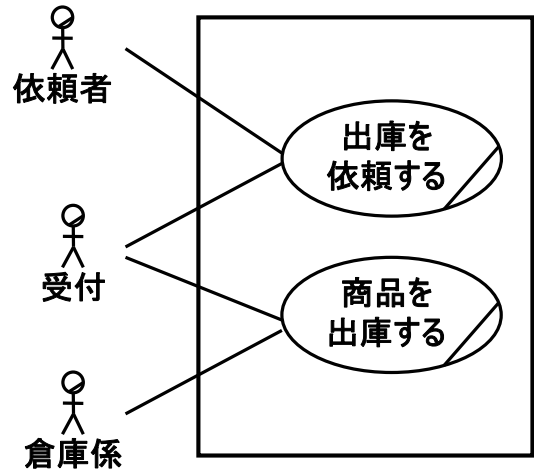
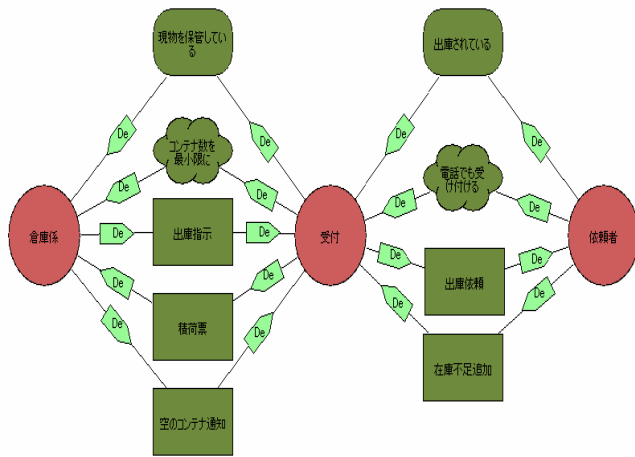
- ユースケース駆動開発の特徴
  - ユースケースは最初に仕様確定する
  - ゴールを管理する
  - 後工程にレビューは存在しない、ユースケース設定ミスは致命的となる



Rational Unified Processに於いて  
ユースケース自体は反復されない、推敲しない、ゴールだから。

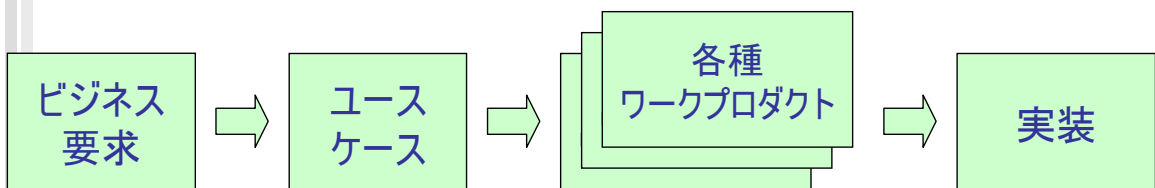
Dependencyモデル

ユースケース図



※法だけが示せるもの：  
 ゴール、アクター間利害関係、  
 依存しているリソース、どのようにやるか(how)

開発プロセスで記述されるもの



形容詞

最新の



名詞

株価情報を

副詞

確実に



動詞

閲覧する



株価情報を  
閲覧する



Short running  
Processで



WS-Reliable  
Messagingで

該当ワークプロダクトなし

非機能要件

現行の開発プロセスでは、形容詞、副詞は視覚化されない

| 品詞        | 対象領域   | 設計 (どのように視覚化するか)   |
|-----------|--|--|
| 名詞的<br>世界 | ◆情報の形、<br>関連性を定義する   | <ul style="list-style-type: none"> <li>●ER図</li> <li>●クラス図</li> <li>●パッケージ図</li> </ul>                                   |
| 動詞的<br>世界 | <ul style="list-style-type: none"> <li>◆インタラクション</li> <li>◆二者間の作用/披作用</li> </ul> | <ul style="list-style-type: none"> <li>●BPEL, BPMN, DFD図</li> <li>●コラボレーション図</li> <li>●シーケンス図</li> </ul>                 |
| 副詞的<br>世界 | <ul style="list-style-type: none"> <li>◆状況</li> <li>◆横断的関心事</li> </ul>           | <p>ココの可視化が難しい</p> <p style="border: 1px solid black; border-radius: 15px; padding: 5px; display: inline-block;">i*法で</p> |

## 元々のビジネス要求

- 早期立ち上げ  
「とにかく早く立ち上げなさい。」
- 高付加価値  
「満足以上のサービスを提供しなさい。」
- 全体ビジネスプロセスの効率化  
「業務の流れの効率化に努めなさい。」

いいものを早く安く提供しなさい。  
 もちろん無駄なことはしては駄目

本来、副詞は重要！：早く、安く、効率的に、無駄なく

## 非機能要件のおさらい、多様なまとめ方の一つ

| 品質特性                | 定義  |
|---------------------|---|
| 機能 Functionality    | 機能の集合の存在及びそれらの明示された性質の存在をもたらす属性の集合。機能は、明示的又は暗示的な必要性を満たすものとする。 |
| 信頼性 Reliability     | 明示された条件の下で、明示された期間、ソフトウェアの達成のレベルを維持するソフトウェアの能力をもたらす属性の集合      |
| 使用性 Usability       | 明示的又は暗示的な利用者の集合が、使用するために必要とする労力及び個々の使用結果による評価に影響する属性の集合       |
| 効率性 Efficiency      | 明示的な条件の下で、ソフトウェアの達成のレベルと使用する資源の量との間の関係に影響する属性の集合              |
| 保守性 Maintainability | 仕様化された改訂を行うために必要な労力に影響する属性の集合                                 |
| 移植性 Portability     | ソフトウェアをある環境から他の環境へ移す際のそのソフトウェアの能力をもたらす属性の集合品質特性               |

## -ilities 31種 (非機能要件)

- accessibility
- accountability
- accuracy
- adaptability
- administrability
- affordability
- agility
- auditability
- availability
- credibility
- standards compliance
- process capabilities
- compatibility
- composability
- configurability
- Correctness
- customizability
- degradability
- demonstrability
- dependability
- deployability
- distributability
- durability
- evolvability
- extensibility
- fidelity
- flexibility
- installability
- Integrity
- interchangeability
- interoperability
- learnability
- maintainability
- manageability
- mobility
- modularity
- nomadicity
- operability
- portability
- precision
- predictability
- recoverability
- relevance
- reliability
- repeatability
- reproducibility
- responsiveness
- reusability
- robustness
- safety
- scalability
- seamlessness
- serviceability (a.k.a. supportability)
- securability
- simplicity
- stability
- survivability
- sustainability
- tailorability
- testability
- timeliness
- understandability
- usability

# 非機能要件 = 副詞



| WS-*                 | 副詞                                 | 動詞   |
|----------------------|------------------------------------|------|
| WS-Security          | 内容が第三者から見えないように、<br>受取人が改ざんできないように | 包む   |
| WS-ReliableMessaging | 到着通知付き送信で                          | 送る   |
| WS-Federation        | 複数の承認経路で                           | 認可する |
| WS-AtomicTransaction | 処理した、しなかったを、確定的に                   | 実行する |
| ...                  | ...                                | ...  |

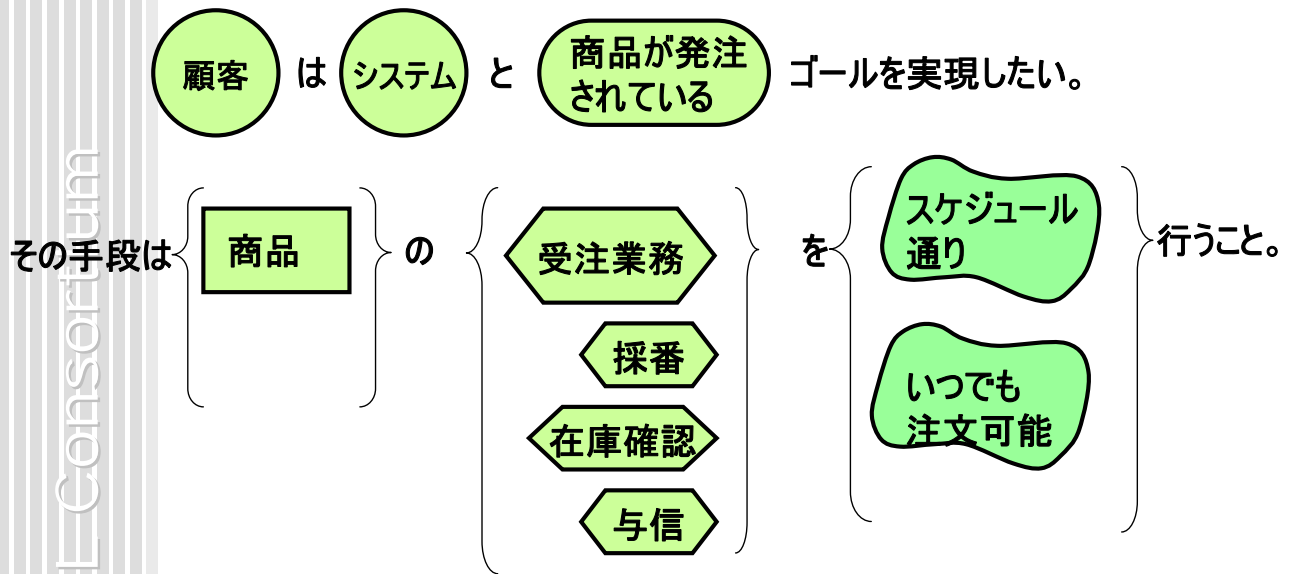
## i\*法でSOAの実装設計の精度向上を狙う

# ソフトゴールを標準化したとすると

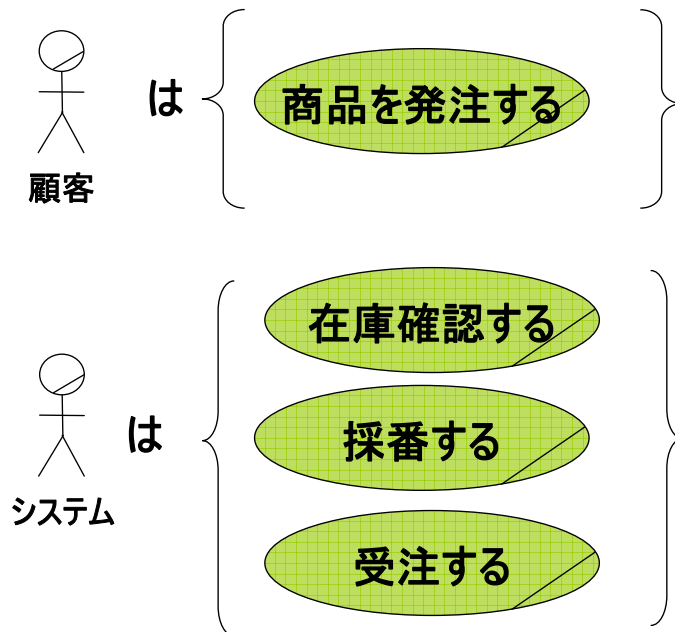


| 非機能要件                | ソフトゴール        |
|----------------------|---------------|
| accessibility        | 誰でも使いやすく      |
| accountability       | 内容証明付きで       |
| accuracy             | 高い精度で         |
| adaptability         | いろいろなものと適合可能で |
| administrability     | 登録しやすく        |
| agility              | 迅速に           |
| auditability         | 対監査性がある       |
| Standards compliance | 標準技術で作ってあり    |
| composability        | 複合組み立て可能で     |
| configurability      | 定義可能で         |
| deployability        | 配布可能で         |
| ...                  | ...           |

## i\*法から要求文を再生成する



## ユースケースを再生成する



アクター間の連携が可視化できない  
非機能要件は書けない

# サービス記述



## サービス設計のアウトライン例

| 項目     | 内容                  |
|--------|---------------------|
| サービス名  | サービスを識別する名前         |
| 内容     | サービスの概要             |
| 利用者の目的 | 利用者がサービスを利用する目的     |
| 提供者の責務 | 提供者が負う責任や範囲         |
| 要望・制約  | 利用者の要望と制約条件         |
| 利用者    | サービスの利用者            |
| 提供者    | サービスの提供者            |
| 入力     | サービス利用者が提供者に与えるリソース |
| 出力     | サービス提供者が利用者を与えるリソース |
| 非機能要件  |                     |

# サービス設計の補強



## 現行のサービス設計書

| 項目     | 内容                  |
|--------|---------------------|
| サービス名  | サービスを識別する名前         |
| 内容     | サービスの概要             |
| 利用者の目的 | 利用者がサービスを利用する目的     |
| 提供者の責務 | 提供者が負う責任や範囲         |
| 要望・制約  | 利用者の要望と制約条件         |
| 利用者    | サービスの利用者            |
| 提供者    | サービスの提供者            |
| 入力     | サービス利用者が提供者に与えるリソース |
| 出力     | サービス提供者が利用者を与えるリソース |
| 非機能要件  | ---フリーフォーマット---     |

## i\* 併用版

| 項目     | 内容   |
|--------|--|
| サービス名  | サービスを識別する名前  |
| 内容     | サービスの概要  |
| 利用者の目的 | 利用者がサービスを利用する目的  |
| 提供者の責務 | 提供者が負う責任や範囲  |
| 要望・制約  | 利用者の要望と制約条件  |
| 利用者    | サービスの利用者   |
| 提供者    | サービスの提供者   |
| 入力     | サービス利用者が提供者に与えるリソース  |
| 出力     | サービス提供者が利用者を与えるリソース  |
| 非機能要件  | <input type="checkbox"/> WS-ReliableMessagingで<br><input type="checkbox"/> WS-Securityで<br><input checked="" type="checkbox"/> BPELで<br><input type="checkbox"/> ESBで<br><input checked="" type="checkbox"/> SCAで<br><input type="checkbox"/> 非同期で<br>:<br>: |

## 非機能要件欄の テンプレート化の可能性

- ユースケースの補完が可能
- SOA = 非機能要件が多い
- 非機能要件管理精度向上メソッドとして i\*
- ソフトゴールの標準化による均質化推進



(意図的に空白)





## ゴール指向分析を用いた サービス候補の導出



### はじめに

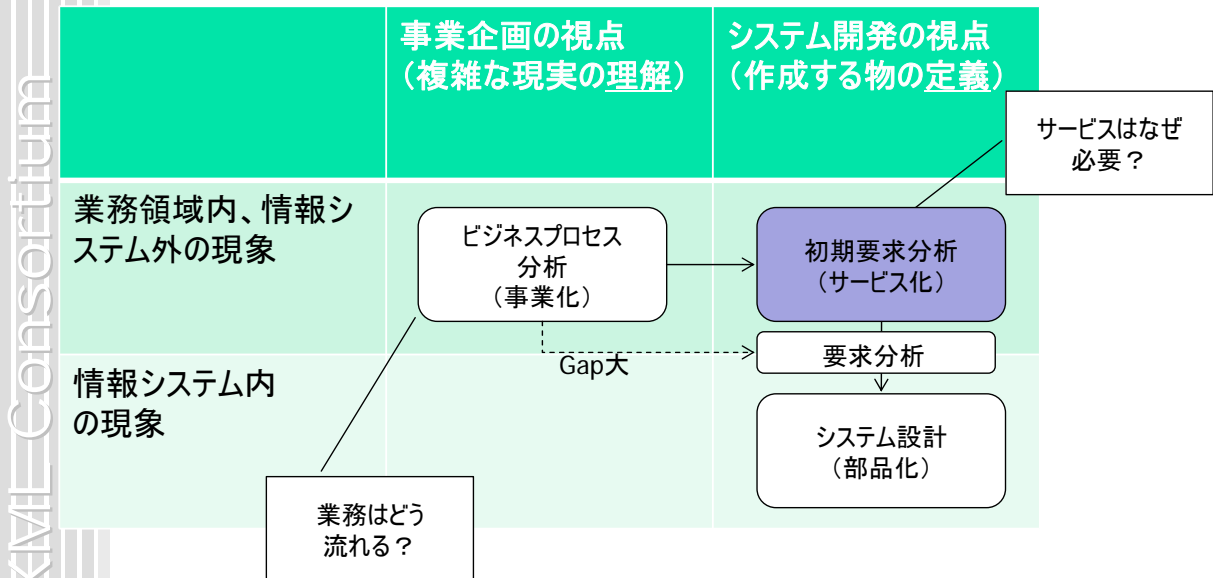


- ビジネスプロセス策定～サービス候補抽出まで、モデリングの思考実験を実施した。
- ビジネス戦略に基づくサービスの導出が課題となった。
- 今回、「ゴール指向分析手法は導出するサービスの有効性に根拠を与える」という仮説を立て検証を行っている。
- i\*法を用いた検証作業から「本当に必要とされるものを提供する」「意味のないシステムを作らない」という可能性は見てきた。

# サービス化の手法



「サービスはなぜ必要？」を考えること  
 ← ゴール指向分析手法で考える



# ワークフロー・モデルを軸にした ビジネスプロセス分析の課題



## 課題

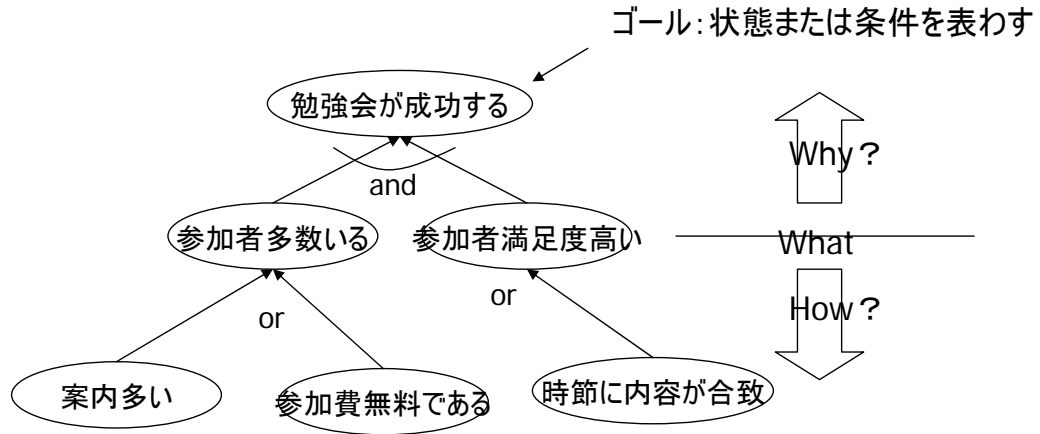
- A. サービス導出の基底となるアクティビティは不安定
- B. サービスの目的が明確にならない
- C. 誤った共通化によるサービス利便性の低下
- D. 人と情報システムの関連が俯瞰できない

# ゴール指向分析

一つの目標(ゴール)を満たすためのより小さい目標に分解する

and結合: 下位全てのサブゴールの達成が必要

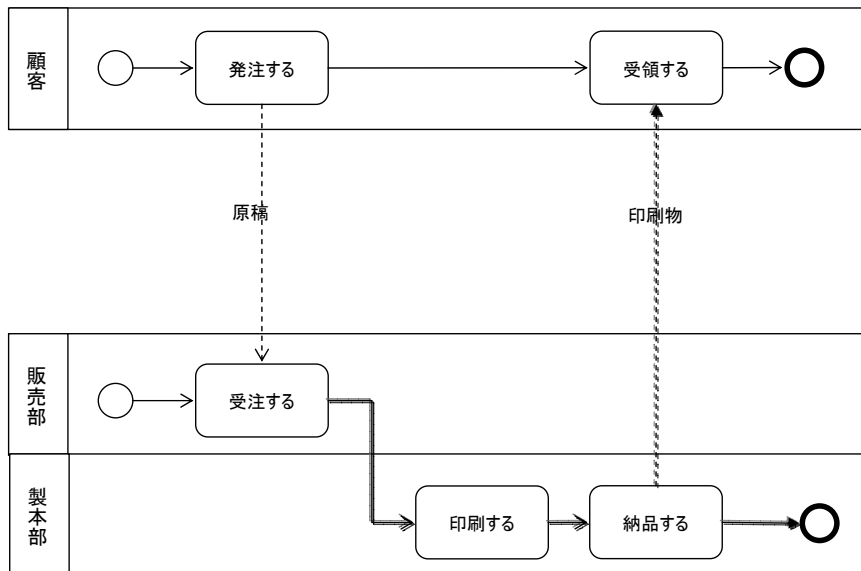
or結合: 下位いずれかのサブゴールの達成が必要



目標を構造的に明らかにすればシステムへの要求が明らかになる

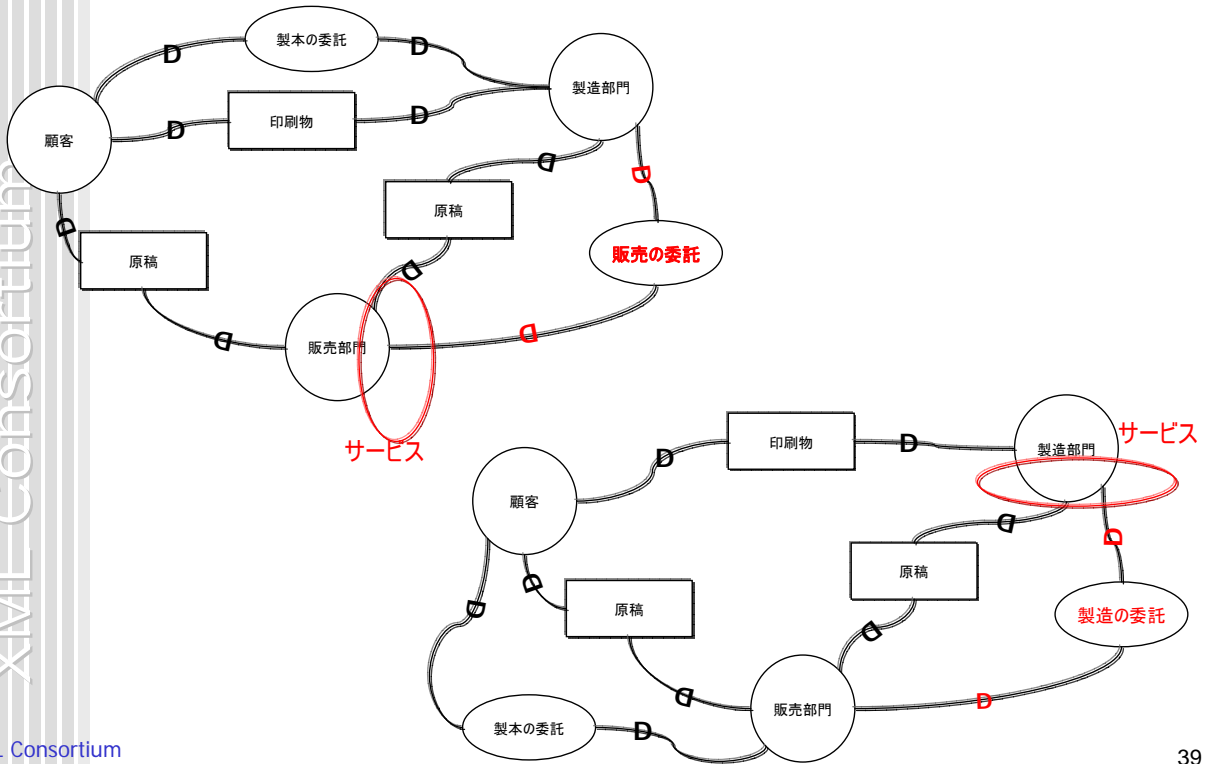
# ワークフロー・モデル

ワークフロー・モデルは事物の因果関係 (Aが起きたらBが起きる) を表現する

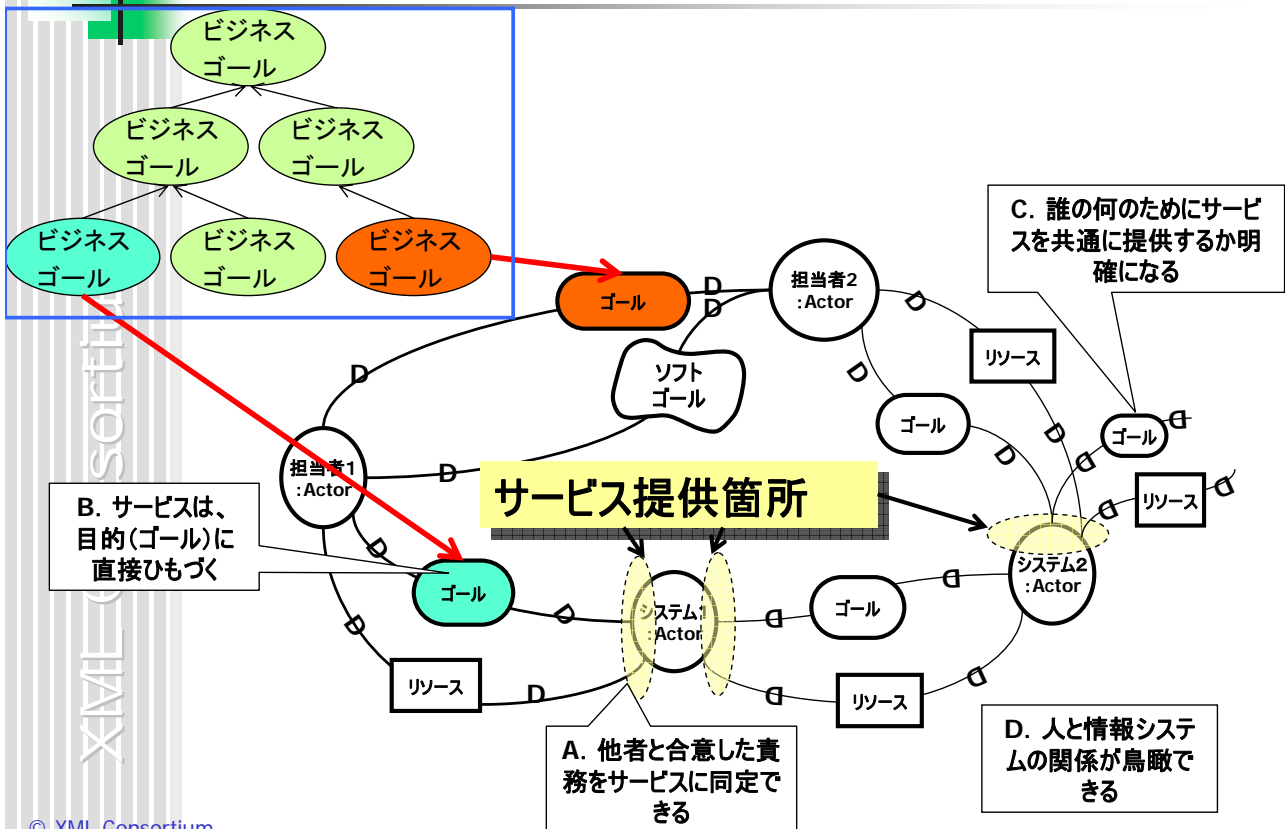


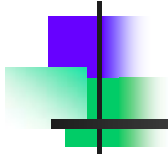
# ゴール・モデル(i\*法)

ゴール・モデルは事物が関係する事由を表現する



# i\*法による課題解決





## 検証結果

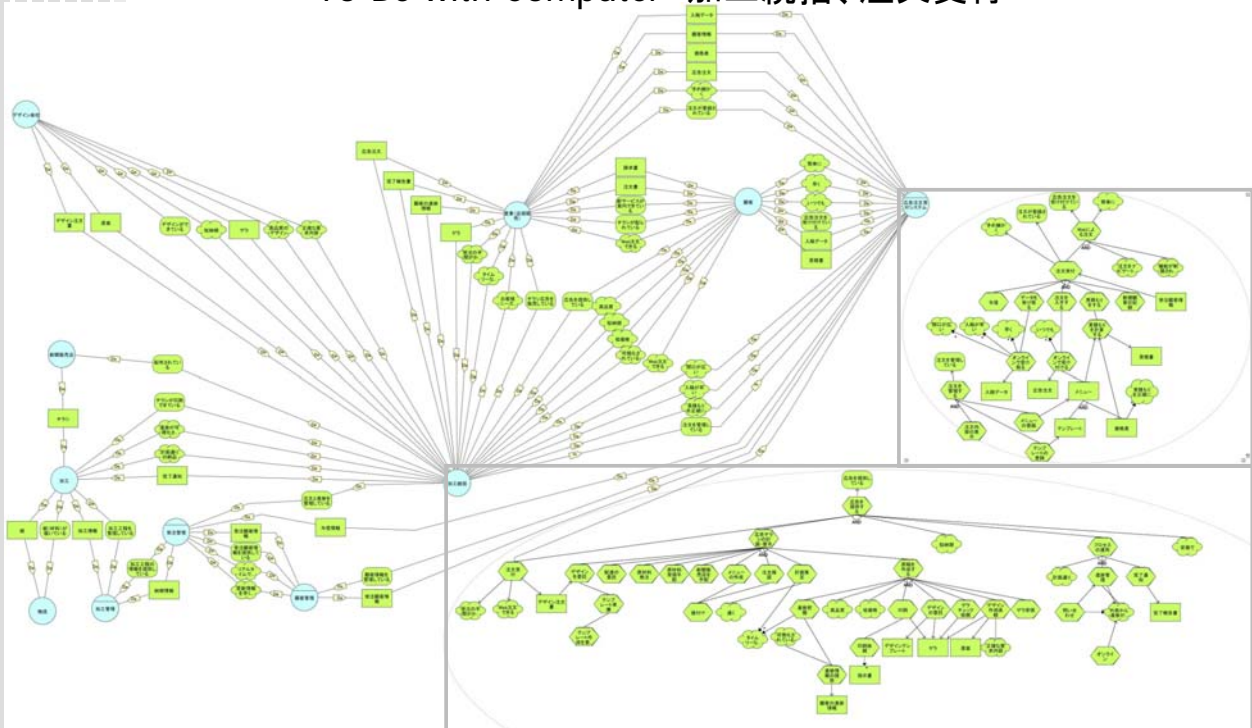


## 検証結果



| 課題                         | 解決できた？   |
|----------------------------|--|
| A. サービス導出の基底となるアクティビティは不安定 | ○ Dependency ゴールと結線しているタスクになる。Dependency ゴールが何か理解するのが難しい。ゴールには複数の階層があり、どの階層のゴールにサービスを合わせるのか判断する必要があるが、情報システムの開発者にはこの判断はできない。 |
| B. サービスの目的が明確にならない         | ○ Dependency ゴールがサービスの目的として明確にできることが確認できた。   |
| C. 誤った共通化によるサービス利便性の低下     | ○ 利用者の目的の類似性で見た場合、同じ目的で利用するサービスは共通化できる。さらにRationaleモデルを比較することで共通化が可能かどうかを検討することができた。   |
| D. 人と情報システムの関連が鳥瞰できない      | ○ Dependencyモデルが俯瞰図である。  |

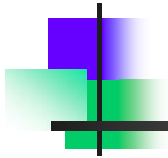
## To-Be with Computer 加工統括、注文受付



# サービス記述

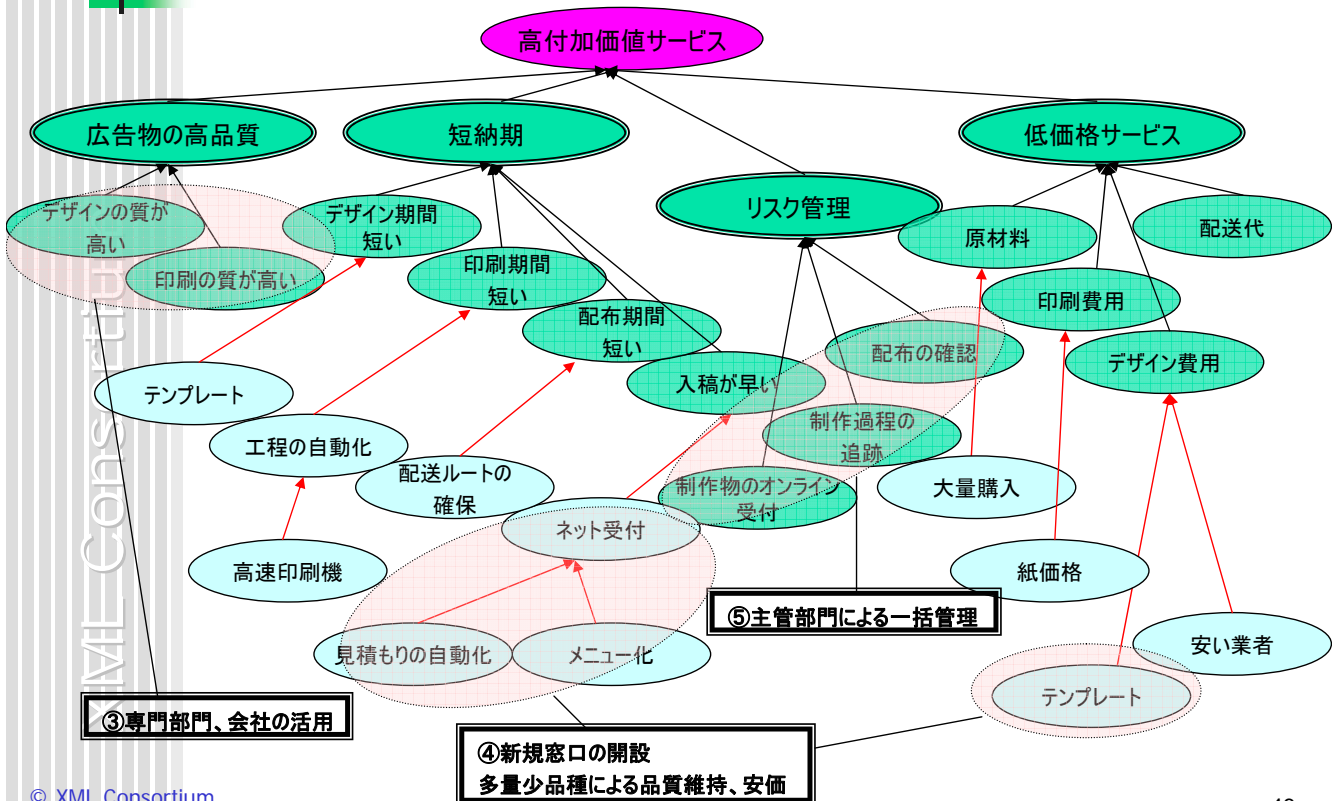
XML Consortium

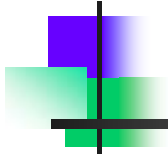
| 項目      | 内容                                    | ビジネスプロセス分析で抽出可・不可 | どこから導出されるのか                          |
|---------|---------------------------------------|-------------------|--------------------------------------|
| サービス名   | Web広告受付サービス                           | ○                 | Dependency ゴールと結線しているタスクの名前 + ソフトゴール |
| サービスの内容 | 広告サービスの受注(内容の選択、見積もり、注文受け付け)をオンラインで行う | △                 | "サービス名"となるタスク配下のタスクとリソース             |
| 利用者の目的  | 広告サービスの発注を行う                          | △                 | Dependency ゴールとリソース                  |
| 提供者の責務  | 広告注文を受け付ける<br>見積もりを正確に行う              | △                 | 提供側の内部ゴール                            |
| 要望・制約   | 随時注文を受け付ける<br>操作が容易                   | △                 | Dependency ソフトゴール                    |
| 利用者     | 顧客                                    | ○                 | アクター名                                |
| 提供者     | 広告管理システム                              | ○                 | アクター名                                |
| 入出力情報   | (入力)顧客情報、入稿データ、<br>(出力)見積書、広告注文       | ○                 | Dependency リソース                      |



## 想定したビジネス戦略

## ビジネス戦略を表したゴール・モデル





## 課題の解決

### ワークフロー・モデルを軸にした ビジネスプロセス分析の課題



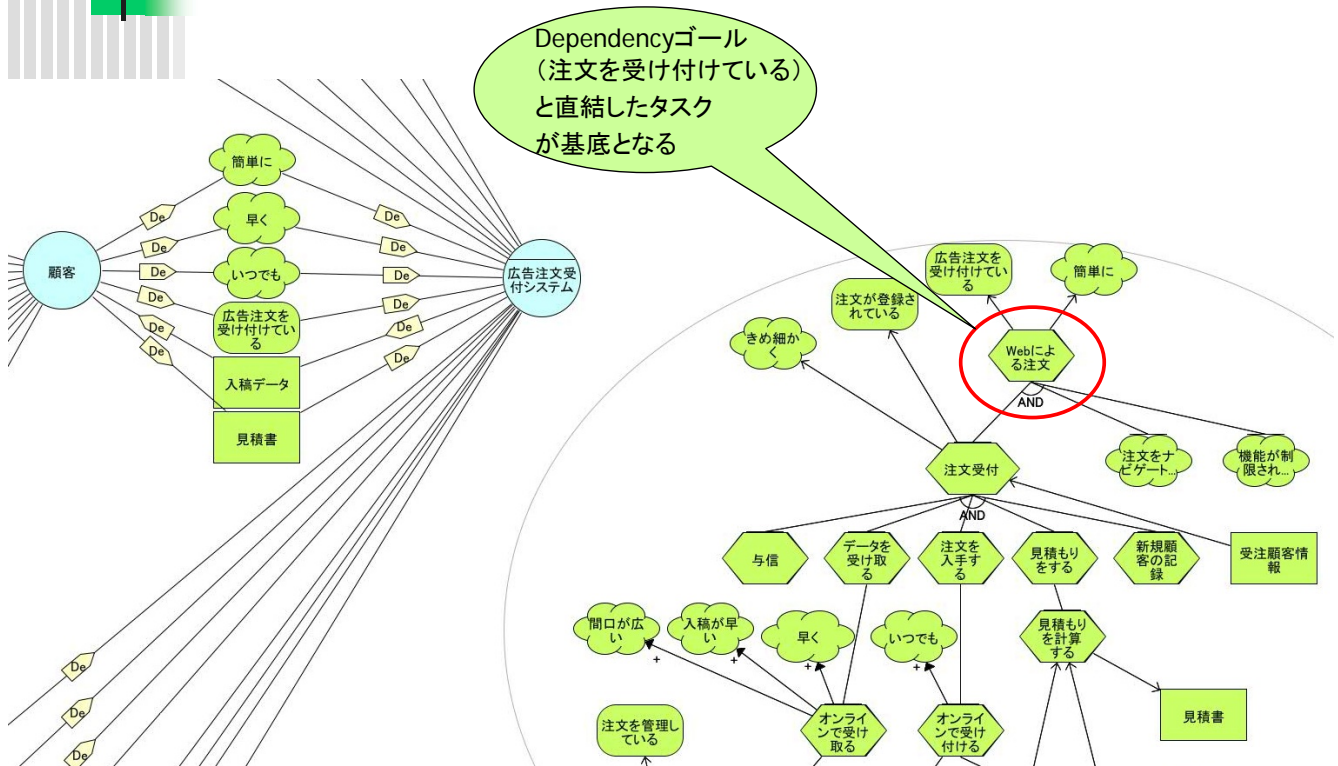
#### 課題

- A. サービス導出の基底となるアクティビティは不安定
- B. サービスの目的が明確にならない
- C. 誤った共通化によるサービス利便性の低下
- D. 人と情報システムの関連が俯瞰できない



# 課題Aの解決

サービス導出の基底となるアクティビティは不安定



## ワークフロー・モデルを軸にした ビジネスプロセス分析の課題



### 課題

A. サービス導出の基底となるアクティビティは不安定

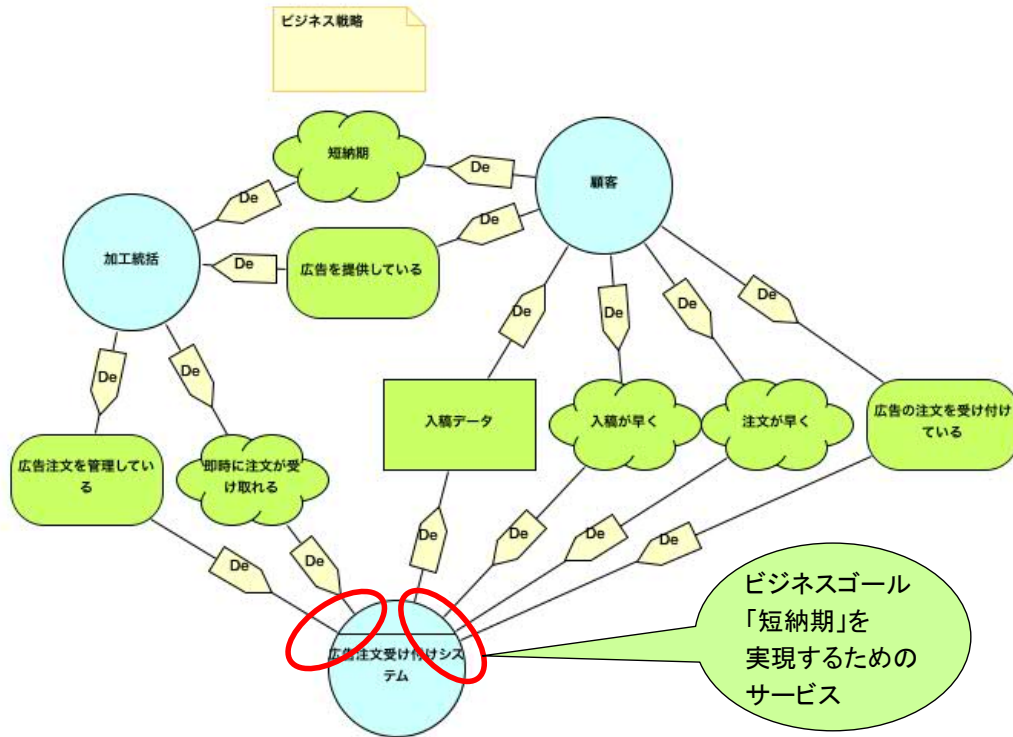
B. サービスの目的が明確にならない

C. 誤った共通化によるサービス利便性の低下

D. 人と情報システムの関連が俯瞰できない

# 課題Bの解決

サービスの目的が明確にならない



# ワークフロー・モデルを軸にした ビジネスプロセス分析の課題



## 課題

A. サービス導出の基底となるアクティビティは不安定

B. サービスの目的が明確にならない

C. 誤った共通化によるサービス利便性の低下

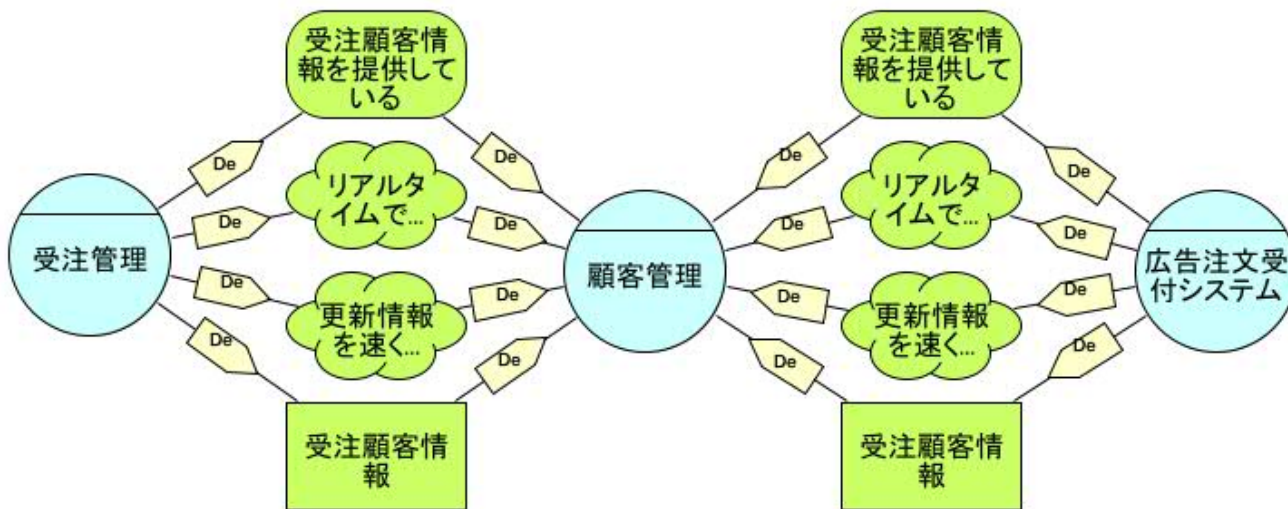
D. 人と情報システムの関連が俯瞰できない

# 課題Cの解決

誤った共通化によるサービス利便性の低下

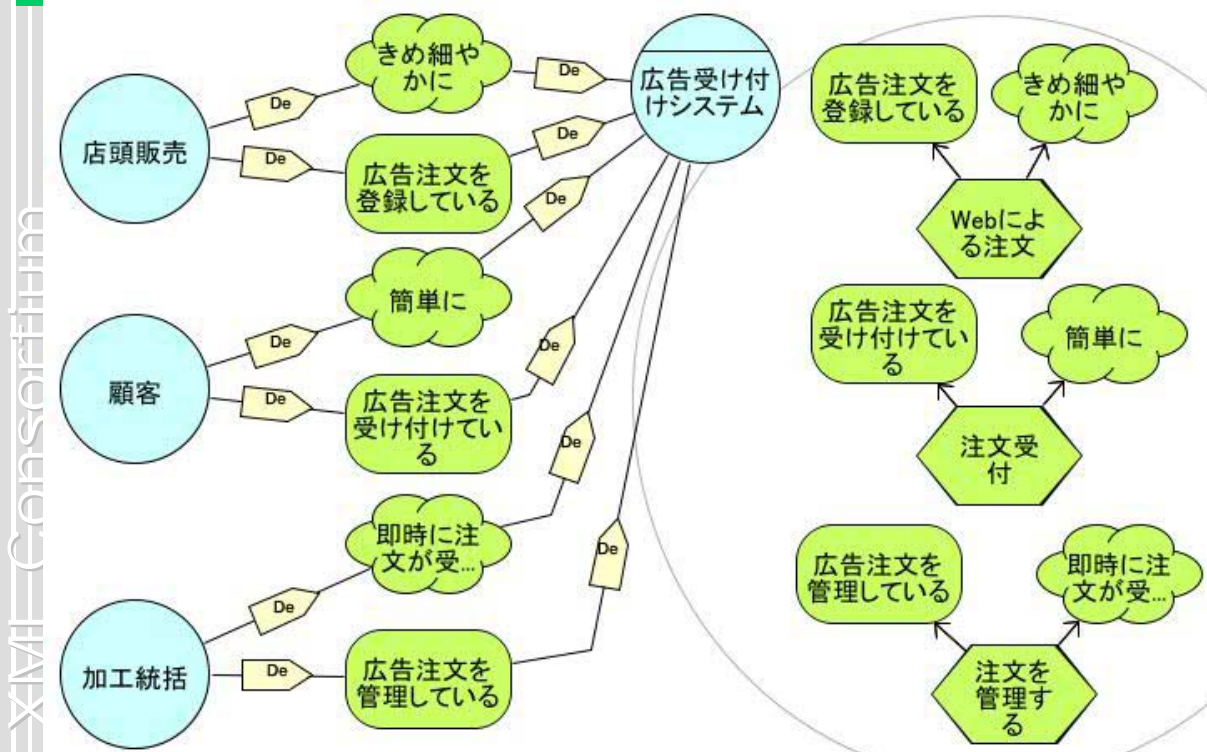


利用者から見たゴール、ソフトゴール、リソースが同じなら共通化出来る



# 課題Cの解決

誤った共通化によるサービス利便性の低下



## 課題

A. サービス導出の基底となるアクティビティは不安定

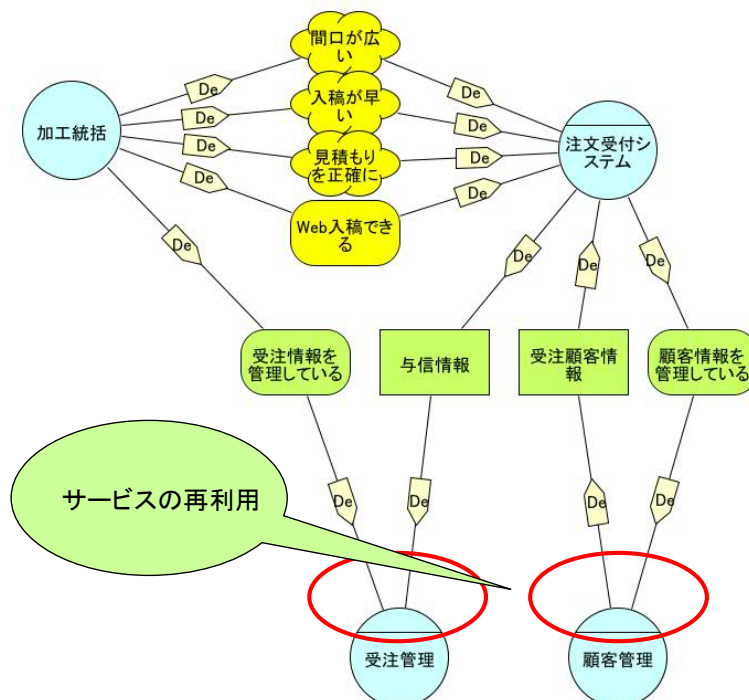
B. サービスの目的が明確にならない

C. 誤った共通化によるサービス利便性の低下

D. 人と情報システムの関連が俯瞰できない

## 人と情報システムの関係 (D)

### システム間の俯瞰図(一部を省略)



| 項目      | 内容                                    | ビジネスプロセス分析で抽出可・不可 | どこから導出されるか                         |
|---------|---------------------------------------|-------------------|------------------------------------|
| サービス名   | Web広告受付サービス                           | ○                 | Dependency ゴールと結線しているタスクの名前+ソフトゴール |
| サービスの内容 | 広告サービスの受注(内容の選択、見積もり、注文受け付け)をオンラインで行う | △                 | “サービス名”となるタスク配下のタスクとリソース           |
| 利用者の目的  | 広告サービスの発注を行う                          | △                 | Dependency ゴールとリソース                |
| 提供者の責務  | 広告注文を受け付ける<br>見積もりを正確に行う              | △                 | 提供側の内部ゴール                          |
| 要望・制約   | 随時注文を受け付ける<br>操作が容易                   | △                 | Dependency ソフトゴール                  |
| 利用者     | 顧客                                    | ○                 | アクター名                              |
| 提供者     | 広告管理システム                              | ○                 | アクター名                              |
| 入出力情報   | (入力)顧客情報、入稿データ、<br>(出力)見積書、広告注文       | ○                 | Dependency リソース                    |

## 仮説検証のまとめ

- サービスはなぜ必要かを説明できる
  - 根拠をモデルとして表すことができる
- 利用者の視点でサービスを導出できる
- ビジネス戦略に基づくSOAベースの情報システム開発に効果がある
  - ビジネス戦略とサービスの目的が関連づけられる