



## i\*法によるサービス分析の試み

2008年 6月4日

XMLコンソーシアム ビジネス・イノベーション研究部会

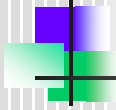
日本ユニシス株式会社 牧野友紀

日本電気株式会社 柴田昌宏

## メンバー一覧



- メンバー(五十音順)
  - 芦田尚人 プレイネットワークス
  - 尾々野正和 日立ソフト(株)
  - 何翁径迪 アイ・ティ・フロンティア(株)
  - 加藤昌宏 住友セメントシステム開発(株)
  - 坂下秀彦 (株)JIEC
  - 柴田昌宏 NEC (株)
  - 谷口伸一 (株)日立製作所
  - 根本和郎 日本IBM(株)
  - 日力俊彦 日本IBM(株)
  - 牧野友紀 日本ユニシス(株)
  - 横田治樹 NECソフト(株)

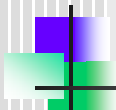


# はじめに



XML Consortium

- ビジネスイノベーション研究部会  
ビジネスプロセスマネジメント SOAシステム構築  
を具体的に検討。
- 大きな課題  
同定サービスの利便性、再利用性の根拠  
原因 = サービス利用者の視点不足
- 今回の取り組み  
ゴール指向分析手法<sup>1</sup>\*法を適用  
「なぜ利用者はそのサービスを必要とするか」
- 研究半ばの内容を報告



# 動機



XML Consortium

ビジネスマネジメント手法BPMの  
ビジネスプロセス・モデル作成

システム化範囲、サービス導出等  
ソフトウェアのモジュール化を検討

一昨年までの活動

ワークフローのアクティ  
ティを基に定義したサービ  
スは、目的が不明確で利  
便性がない

課題

モジュールの合理性が乏  
しく再利用できない可能  
性がある

ゴール指向分析手法を導入することによりビジネス  
プロセス分析からのサービス設計方法を確立する

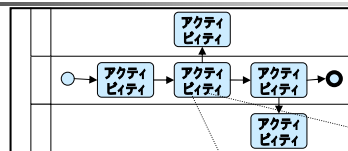
今年の活動

## これまでの活動のおさらい

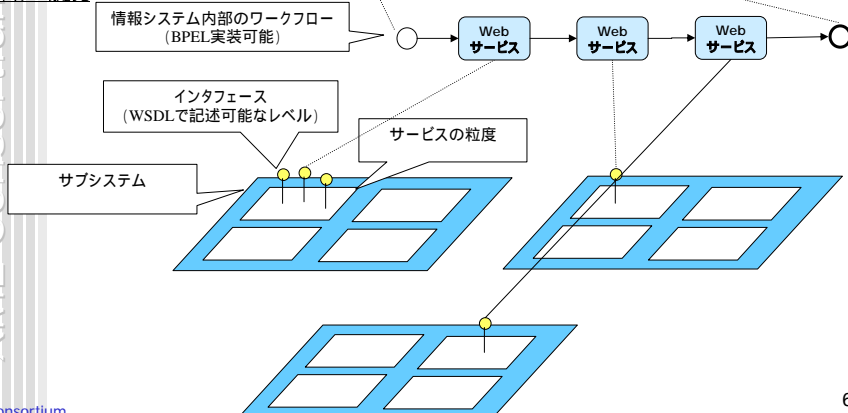
## プロセスと情報システムとの関係



### ビジネスプロセス設計



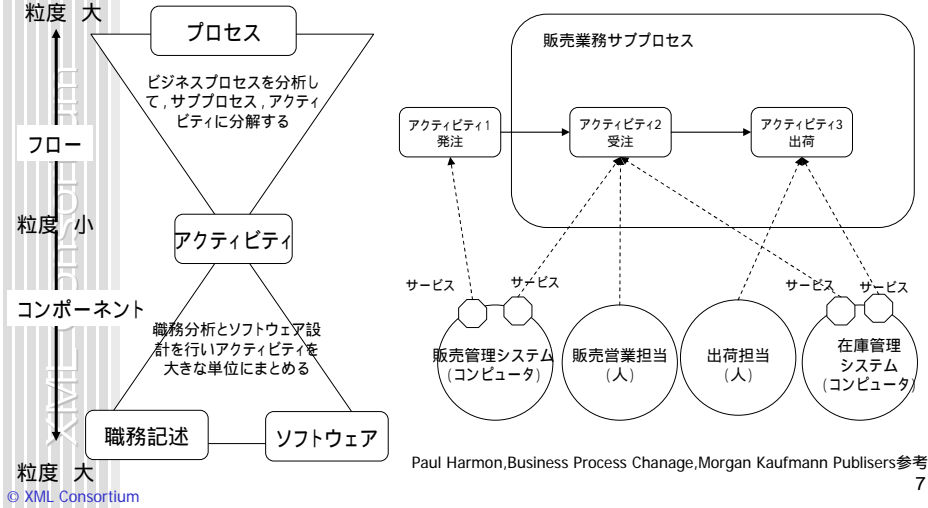
### 情報システム開発



# アクティビティがサービスとの接点



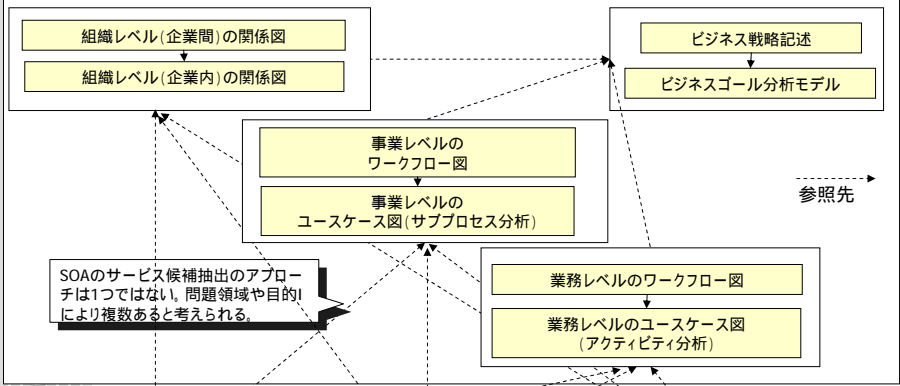
- 「プロセス」と「サービス」の接点はプロセスの1つ1つを構成するアクティビティとなり、アクティビティが利用するサービスのセットが情報システムとして実現されることになります。



# モデル間の関連

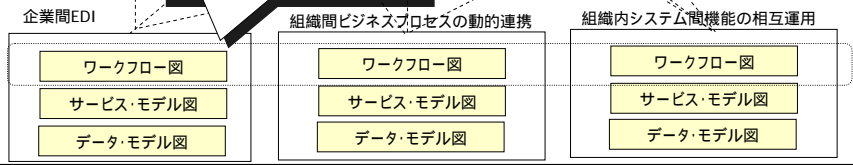


ビジネス領域 (コンピュータ非依存)



IT領域

BPMN or UML->BPELのMDA的な実行可能コード生成の領域



## サービス分析・設計の流れ



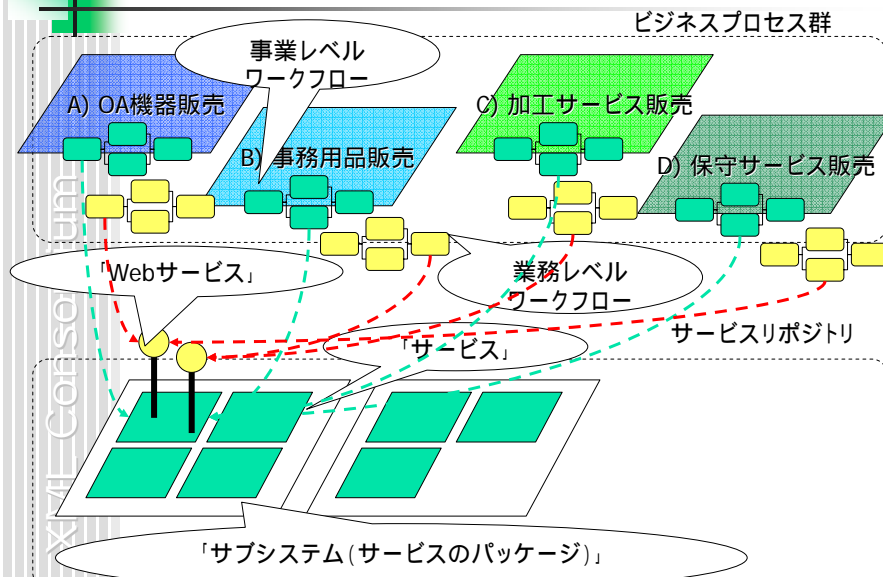
1) 事業レベル・ワークフローの共通するサブプロセス (=サービス) を抽出

2) 共通サブプロセスを分類 (= サブシステム (サービスのパッケージ))

3) 業務レベル・ワークフローの共通するアクティビティ (=Webサービス) を抽出

4) 共通するアクティビティのユースケースを分析して、情報システムが提供する機能 (=Webサービス仕様) を定義

## ビジネスプロセスの共通化と分類

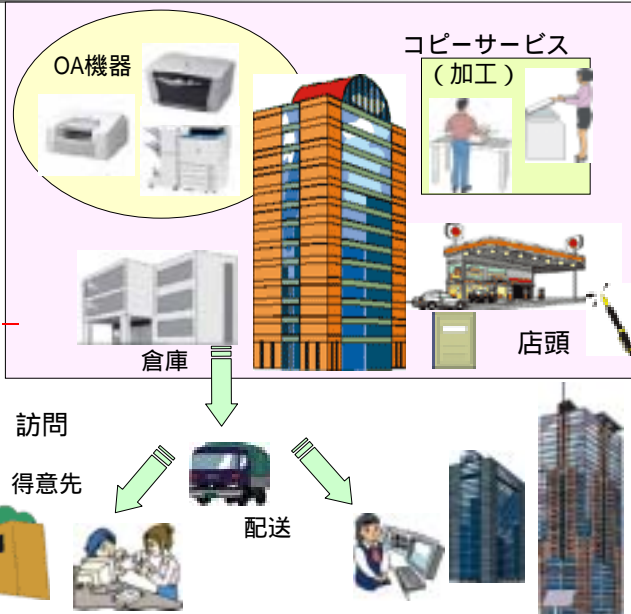


# 業務モデル: OA機器販売業者



XML Consortium

新事業戦略

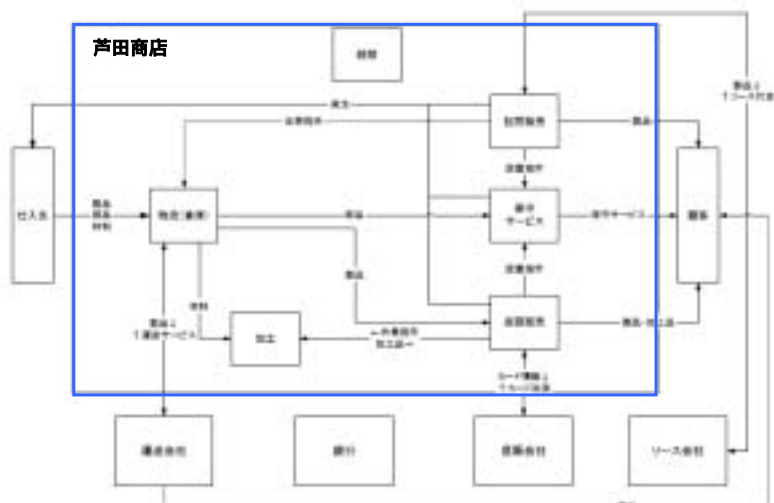


# 組織レベルの関係図

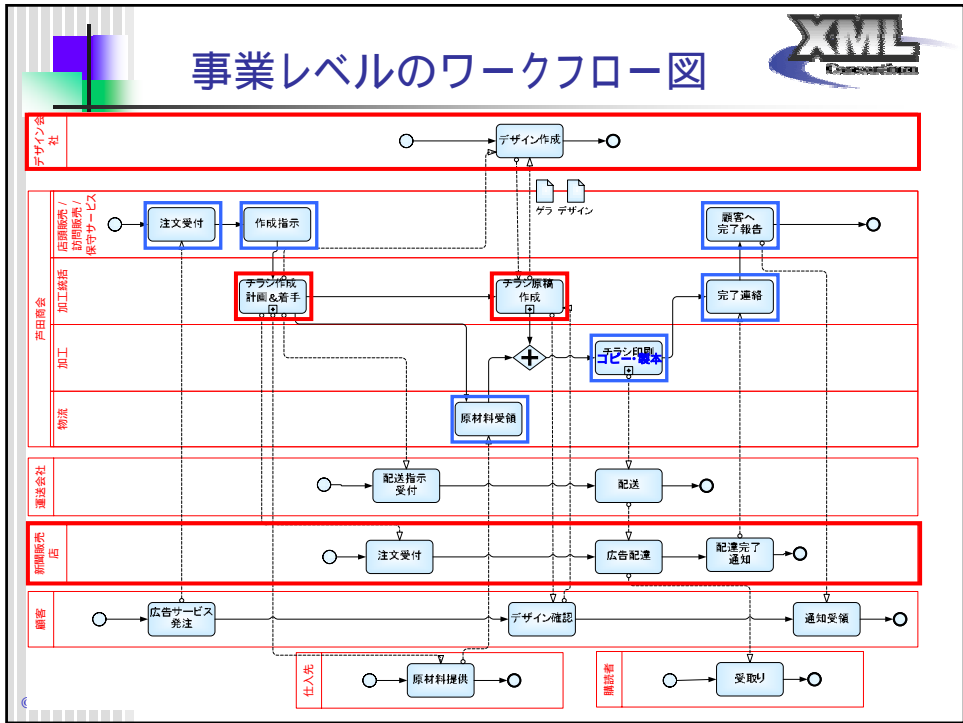


物や資金、情報をやり取りする組織間の関係を明確にする

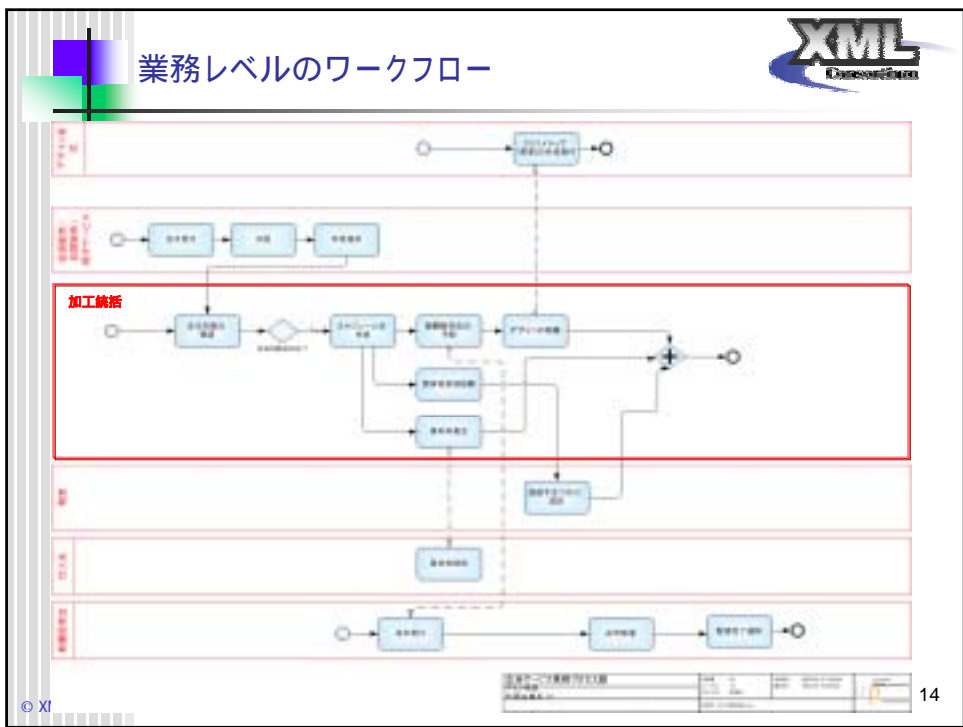
XML Consortium



# 事業レベルのワークフロー図




# 業務レベルのワークフロー



| サービス・パッケージ                | サービス   | 関連サブプロセス  | サービス・パッケージ       | サービス   | 関連サブプロセス   |
|---------------------------|--|---|------------------|--|--|
| 受注処理サービス・パッケージ            | 注文サービス   | <ul style="list-style-type: none"> <li>注文受付サブプロセスの店頭販売業務プロセス</li> <li>注文受付サブプロセスの加工サービス業務プロセス</li> <li>注文受付サブプロセスの保守サービス業務プロセス</li> </ul> | 受注サービス・パッケージ     | 注文受付サービス   | <ul style="list-style-type: none"> <li>注文受付サブプロセスの店頭販売業務プロセス</li> <li>注文受付サブプロセスの事務用品販売業務プロセス</li> <li>注文受付サブプロセスの加工サービス業務プロセス</li> <li>注文受付サブプロセスの保守サービス業務プロセス</li> </ul>                                     |
|                           | 与信サービス   | <ul style="list-style-type: none"> <li>リース会社と与信確認サブプロセスの店頭販売業務プロセス</li> <li>与信サブプロセスの加工サービス業務プロセス</li> </ul>                              |                  | 与信サービス   | <ul style="list-style-type: none"> <li>与信サブプロセスのOA機器販売業務プロセス</li> <li>与信サブプロセスの事務用品販売業務プロセス</li> <li>与信サブプロセスの加工サービス業務プロセス</li> </ul>  |
|                           | 在庫確認サービス   | <ul style="list-style-type: none"> <li>注文受付サブプロセスの店頭販売業務プロセス</li> <li>注文受付サブプロセスの加工サービス業務プロセス</li> <li>注文受付サブプロセスの保守サービス業務プロセス</li> </ul> |                  | 作業指示サービス   | <ul style="list-style-type: none"> <li>出庫指示サブプロセスの店頭販売業務プロセス</li> <li>加工指示サブプロセスの加工サービス業務プロセス</li> <li>保守契約サブプロセスの保守サービス販売</li> </ul>  |
|                           | 作業指示サービス   | <ul style="list-style-type: none"> <li>出庫指示サブプロセスの店頭販売業務プロセス</li> <li>加工指示サブプロセスの加工サービス業務プロセス</li> <li>保守契約サブプロセスの保守サービス販売</li> </ul>     |                  | 設置指示サービス   | <ul style="list-style-type: none"> <li>設置指示サブプロセスのOA機器販売業務プロセス</li> </ul>  |
| 出庫処理サービス・パッケージ            | <ul style="list-style-type: none"> <li>出庫指示サービス</li> <li>設置指示サービス</li> </ul>               | <ul style="list-style-type: none"> <li>出庫指示サブプロセスの店頭販売業務プロセス</li> <li>設置指示サブプロセスの店頭販売業務プロセス</li> </ul>                                    | OA機器販売サービス・パッケージ | <ul style="list-style-type: none"> <li>商品提供サービス</li> <li>加工指示サービス</li> <li>設置作業サービス</li> <li>保守契約サービス</li> </ul> | <ul style="list-style-type: none"> <li>商品提供サブプロセスの事務用品販売業務プロセス</li> <li>加工指示サブプロセスの加工サービス業務プロセス</li> <li>チラシ印刷サブプロセスの広告サービス業務プロセス</li> <li>設置作業サブプロセスのOA機器販売業務プロセス</li> <li>保守契約サブプロセスの保守サービス業務プロセス</li> </ul> |
| 加工処理サービス・パッケージ            | <ul style="list-style-type: none"> <li>コピーサービス</li> <li>製本サービス</li> </ul>                  | <ul style="list-style-type: none"> <li>コピー-製本サブプロセスの加工サービス業務プロセス</li> <li>コピー-製本サブプロセスの加工サービス業務プロセス</li> </ul>                            | 出庫サービス・パッケージ     | 出庫サービス   | <ul style="list-style-type: none"> <li>出庫サブプロセスのOA機器販売業務プロセス</li> </ul>  |
| クリエイティブサービス・パッケージ(広告サービス) | <ul style="list-style-type: none"> <li>デザイン作成サービス</li> <li>印刷依頼サービス</li> </ul>             | <ul style="list-style-type: none"> <li>デザインの依頼サブプロセスのチラシ作成計画&amp;着手サブプロセス</li> <li>新聞配達店の手配サブプロセスのチラシ作成計画&amp;着手サブプロセス</li> </ul>         | コピー-製本サービス・パッケージ | コピー-製本サービス   | <ul style="list-style-type: none"> <li>コピー-製本サブプロセスの加工サービス業務プロセス</li> </ul>  |
| 保守サービス・パッケージ              | <ul style="list-style-type: none"> <li>契約処理サービス</li> <li>設置サービス</li> <li>点検サービス</li> </ul> | <ul style="list-style-type: none"> <li>保守契約サブプロセスの保守サービス業務プロセス</li> <li>設置作業サブプロセスの店頭販売業務プロセス</li> <li>なし</li> </ul>                      | チラシ作成サービス・パッケージ  | <ul style="list-style-type: none"> <li>チラシ作成計画&amp;着手サービス</li> <li>チラシ印刷作成サービス</li> </ul>                        | <ul style="list-style-type: none"> <li>チラシ作成計画&amp;着手サブプロセスの広告サービス業務プロセス</li> <li>チラシ印刷サブプロセスの広告サービス業務プロセス</li> </ul>  |
| 納品処理サービス・パッケージ            | <ul style="list-style-type: none"> <li>配達サービス</li> <li>手渡しサービス</li> </ul>                  | <ul style="list-style-type: none"> <li>配達サブプロセスの店頭販売業務プロセス</li> <li>商品提供プロセスの店頭販売業務プロセス</li> </ul>  |                  |  |  |

© XML Consortium



# XML Consortium

## ビジネスプロセス分析を用いる課題

© XML Consortium



- A) サービス導出の規定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

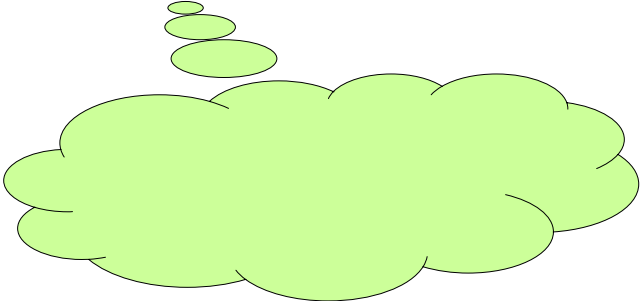
## 課題A

- A) サービス導出の規定となるアクティビティは不安定
  - 分析目的のビジネスプロセス・モデルのアクティビティは不安定。安定的なサービスを同定するには一工夫必要。
  - 現状の業務を理解することを目的にしたビジネスプロセス・モデルは、活動の流れ(因果関係)を大まかに表現する。アクティビティの粒度を厳密に揃えることに意味がない。

分析モデルから設計モデルへ機械的な変換  
はできない。

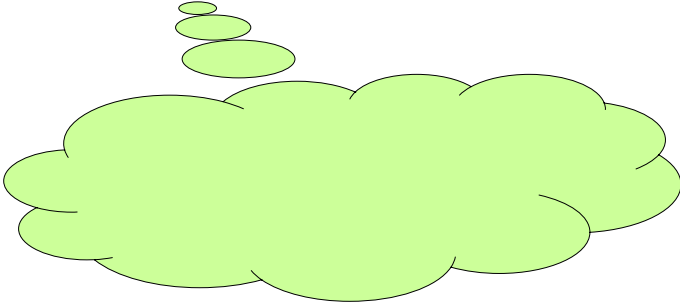
複雑な現実の理解 システムの正確な定義

- B) サービスの目的が明確にならない
- ビジネスプロセス・モデルの個々のアクティビティを、ビジネス上の目的と関連付けていない。 ビジネスプロセスだけでは、サービスの目的は明確にならない。



ビジネスプロセス(因果関係)から、  
サービスを利用する目的を知ることはできない。

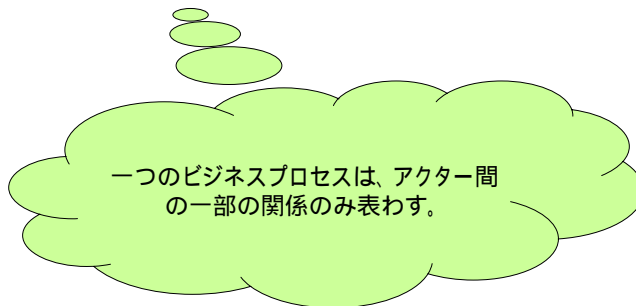
- C) 誤った共通化によるサービス利便性の低下
- 属性の類似度から異なるビジネスプロセスのサブプロセスやアクティビティを共通化する。詳細なレベルの個別性を無視することで利便性が低下したり、時間の経過とともに個別性が大きくなり分割せざるを得ない可能性がある。意義のある共通化のためには、別の視点が必要。



作り手の視点で共通化している  
使い手の視点で見れば、利便性と可変性は見えない

D) 人と情報システムの関連が俯瞰できない

- ビジネスプロセスのアクティビティ毎に、ユースケースを分析することで情報システムと人との関わりが部分的に明らかになるが、全体的な視点で人と情報システムの関係が見えづらい。業務変更による情報システムの影響を驚づかみにできない。

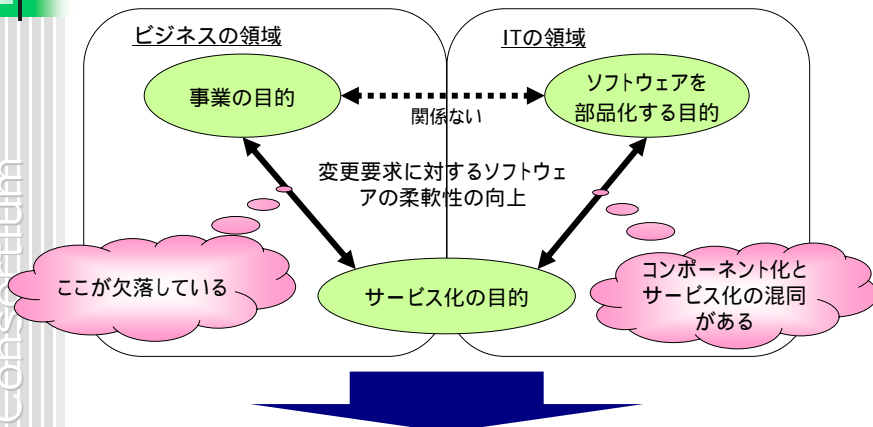
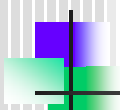


## 役立つサービス分析法プロジェクト



- 目的: 有用なサービスを作る分析の方法を探る
- 手段: i\*法を含めてゴール分析手法の適用
- 成果: サービス分析方法とサンプル・モデル
- 活動内容:
  - サービス分析の方法の仮説
  - 特定業務を想定したモデリングによる思考実験
  - ビジネスプロセス分析に依る方法との比較、相関
- 活動期間: 2007年6月～2008年5月の一年間
- 形態: ビジネス・イノベーション研究部会内の活動

我々の仮説を説明することが目的  
本格的な検証は来年の見込み。



目的と手段の関係を明確にするゴール指向分析手法を導入する。  
ゴール指向分析手法により、情報システムと人の関係、情報システム間関係を明確にする。  
ゴール指向分析手法としてi\*法を用いて考察する。

## i\* 法を検討する動機

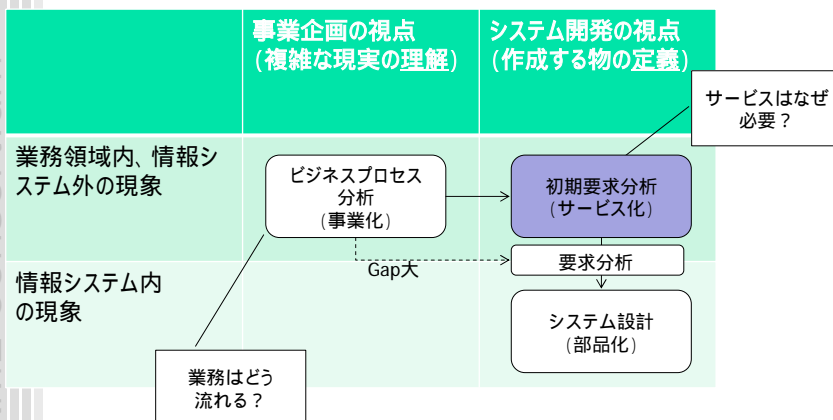


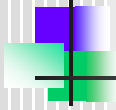
- サービスは現実世界の何を表しているのか
    - (A) *Object Oriented Design works because an Object represents a real-world "thing".*
    - (B) *Service Oriented Architecture works because a Service represents a real world "what we do".*
- STEVE JONES, A METHODOLOGY FOR SERVICE ARCHITECTURES,  
OASIS SOA Adoption Blueprints TC
- ビジネスにおいて“what we do”は“what we do for others”と言えそう。
  - i\* 法は、アクター間の協調関係を目的の視点で明確にする。その目的を達成するために、各アクター内で何を行う必要があるか手段を明確にする。
  - ゆえに、i\* 法は (B) と親和性が高い

## 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





# ゴール指向分析



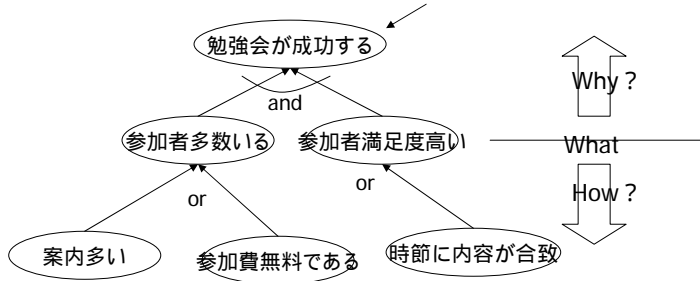
XML Consortium

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

ゴール: 状態または条件を表わす



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

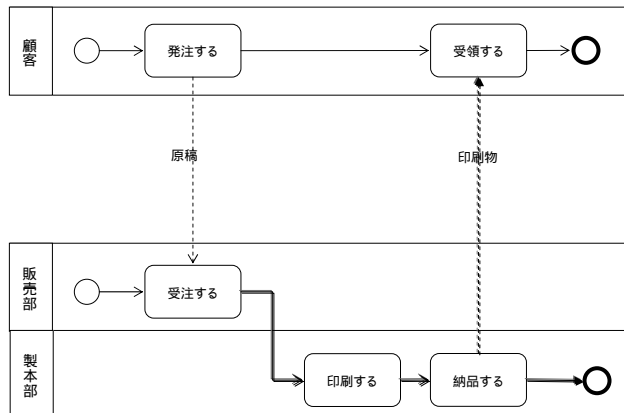


# ワークフロー・モデル



XML Consortium

ワークフロー・モデルは事物の因果関係を表現する

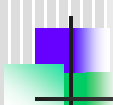
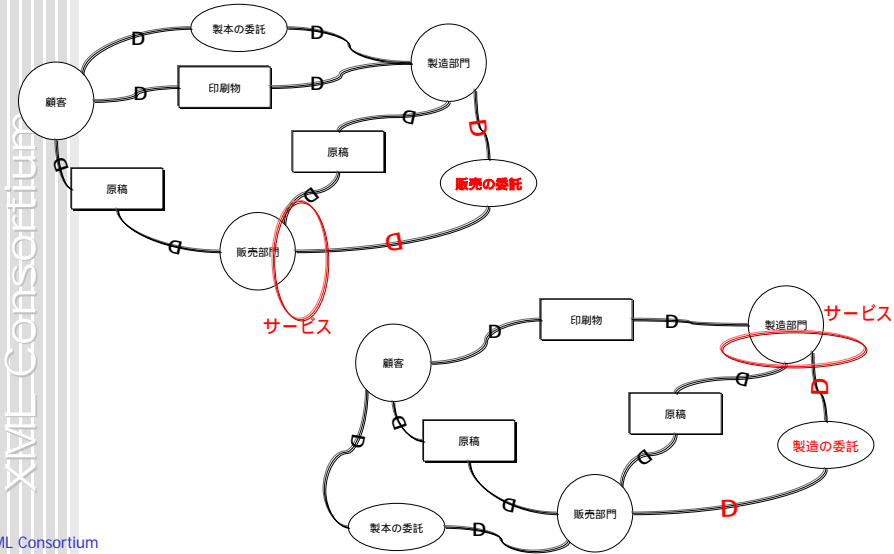




# ゴール・モデル(i\*法)



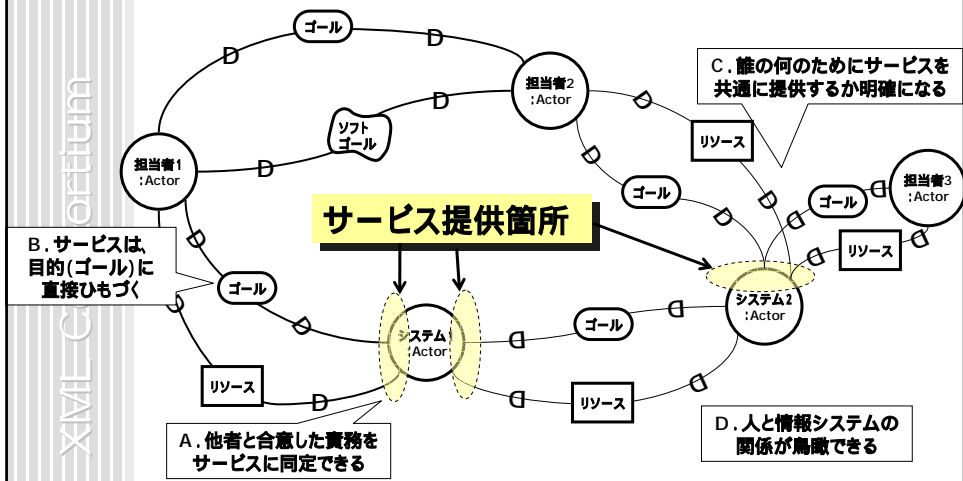
ゴール・モデルは事物が関係する事由を表現する



# i\*法による課題解決



サービスを誰が、誰のために、どんな目的のため、なんの情報を提供するのが明確にする



| 課題                         | 解決策  |
|----------------------------|--|
| A. サービス導出の基底となるアクティビティは不安定 | 協調する目的に着目し、関連するタスクをサービス導出の基底とする              |
| B. サービスの目的が明確にならない         | 協調目的とサービスを関連付ける                              |
| C. 誤った共通化によるサービス利便性の低下     | サービス利用者の視点で目的の共通性と相違性                        |
| D. 人と情報システムの関連が俯瞰できない      | 人と人、人とシステム、システムとシステム間の活動を、目的と手段の観点で関係を、明確にする |

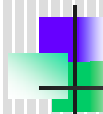
## サービス記述

i \* 法を用いて明らかにするサービスの属性

| 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にするリソース  |



## i\*法の記法



## モデル構成要素



**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.



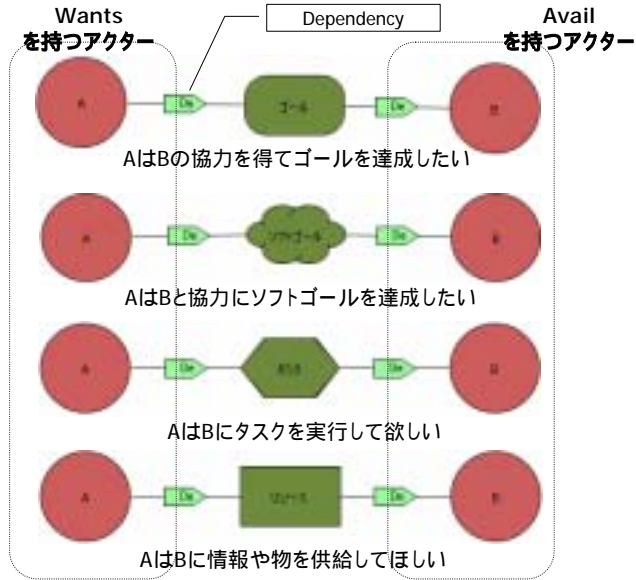
**リソース:** ゴール達成(タスク遂行)に利用する物や情報.



# 依存関係



XML Consortium



© XML Consortium

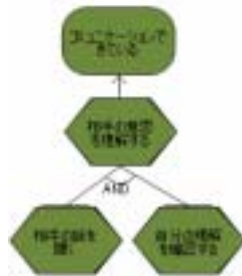


# 構成要素間の関係



XML Consortium

## タスク分割リンク



上位のゴールのタスク分割  
上位のゴールの達成に必要なタスク

## 目的手段(Means-End)リンク



上位のゴールの目的手段展開  
上位のゴールの達成の手段となるタスク

© XML Consortium

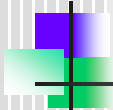
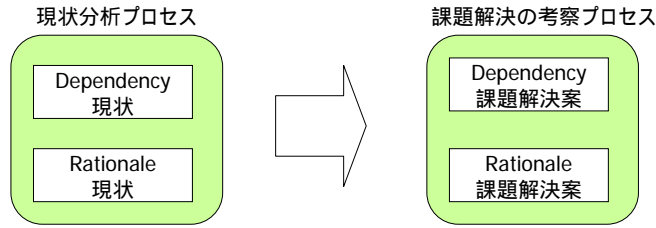


# モデルとモデリング・プロセス



XML Consortium

- 二つのモデル
  - Dependencyモデル(Strategic Dependency Model)
    - アクターの意図を、アクター間の依存関係で表したモデル。
    - 対象とする業務領域の構造を明らかにして、共通のビジネス目標に対して関連するアクター個々の意図がどのように関連(寄与)するか把握するためのモデル
  - Rationaleモデル(Strategic Rationale Model)
    - アクターの意図を実現する、アクター内部の仕組みをタスク等の関係で表したモデル
    - 意図が実現できる根拠を明確にし、また、アクターが考える不都合や要望がどこにあるか明確にするためのモデル
- モデリング・プロセス



# モデリング例



XML Consortium

- 題材(酒屋モデル)
  - 共通問題によるプログラム設計技法解説
    - 情報処理学会 Vol.25 No.91 Sep.1984

ある酒類販売会社の倉庫では、毎日数個のコンテナが搬入されてくる。その内容はビン詰めの酒で、1つのコンテナには10銘柄まで混載できる。扱い銘柄は約200種類ある。倉庫係は、コンテナを受け取りそのまま倉庫に保管し積荷票を受付係へ手渡す。また受付係からの出庫指示によって内蔵品を出庫することになっている。内蔵品は別のコンテナに詰め替えたり、別の場所に保管することはない。空になったコンテナはすぐに搬出される。

積荷票: コンテナ番号(5桁)、搬入年月、日時、内蔵品名、数量(の繰り返し)

さて受付係は毎日数10件の出庫依頼を受け、その都度倉庫係へ出庫指示書を出すことになっている。出庫依頼は出庫依頼票または電話によるものとし、1件の依頼では、1銘柄のみに限られている。在庫が無いが数量が不足の場合には、その旨依頼書に電話連絡し同時に在庫不測リストに記入する。また空になる予定のコンテナを倉庫係に知らせることになっている。倉庫内のコンテナ数はできる限り最小にしたいと考えているからである。

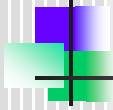
出庫依頼: 品名、数量、送り先名

受付係の仕事(在庫なし連絡、出庫指示書作成および在庫不測リスト作成)のための計算機プログラムを作成せよ。

出庫指示書: 注文番号、送り先名、コンテナ番号、品名、数量、空コンテナ搬出マーク

在庫不測リスト: 送り先名、品名、数量

- ・なお移送や倉庫保管中に酒類の損失は生じない。
- ・この課題は現実的でない部分もあるので、入力データのエラー処理などは省略に扱ってよい。
- ・以上あいまいな点は、適当に解釈してください。



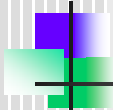
# Dependencyモデル (酒屋モデル)



XML Consortium



© XML Consortium

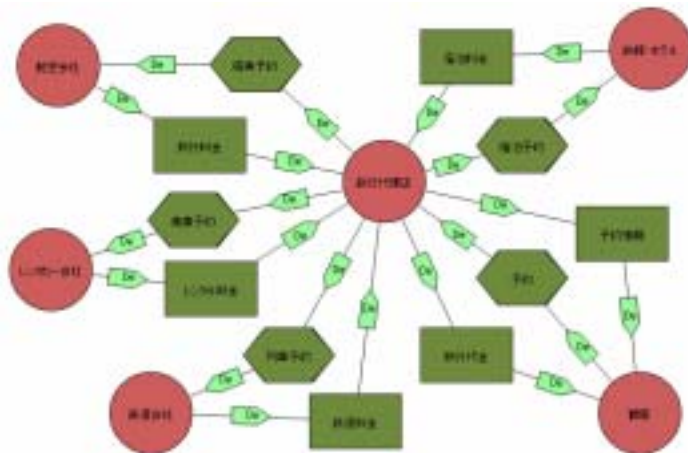


# Dependencyモデル



XML Consortium

- アクター「間」の依存関係を分析します。
- 旅行代理店サービスの例



© XML Consortium

出典: ~ゴール指向による!!~システム要求管理技法 (ソフト・リサーチ・センター, 2007)を元に作成



# Rationaleモデル



- アクター「内」の構成要素を分析します。

旅行代理店サービスの例



© XML Consortium

出典: ~ゴール指向による!!~システム要求管理技法 (ソフト・リサーチ・センター, 2007)を元に作成

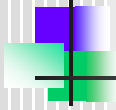


## XML Consortium

### 役立つサービス分析法プロジェクトで導出したサービス



© XML Consortium



# サービス記述例



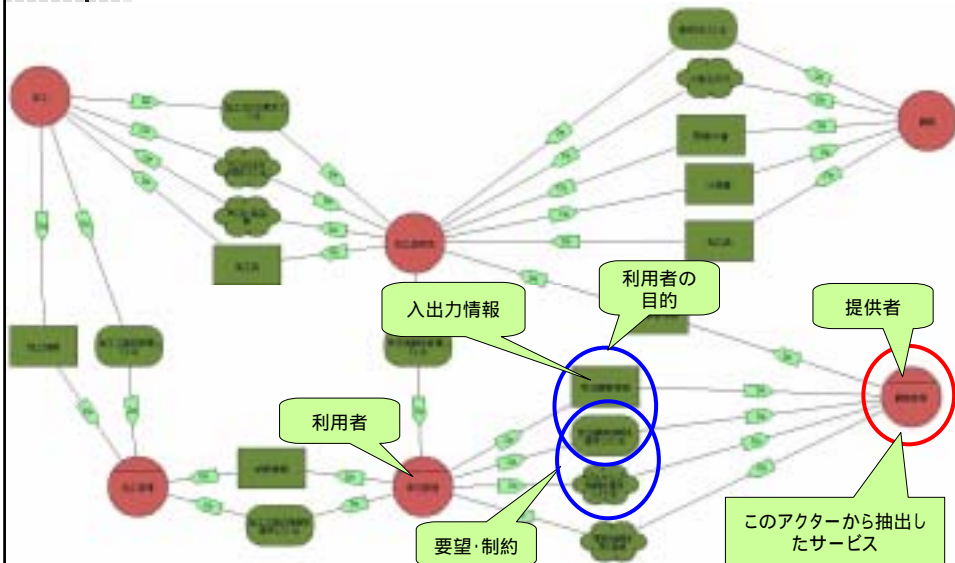
XML Consortium

| 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   | 顧客情報提供サービス                    |                   | Dependency ゴールと結線しているタスクの名前 |
| サービスの内容 | すべての顧客情報から受注顧客情報を検索して依頼者に提供する |                   | “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  | 注文書作成のための必要顧客情報を入手する          | ×                 | Dependency ゴールとリソース         |
| 提供者の責務  | 常に最新の顧客情報を提供する                | ×                 | 提供側の内部ゴール                   |
| 要望・制約   | 顧客情報をリアルタイムに提供する              | ×                 | Dependency ソフトゴール           |
| 利用者     | 受注管理システム                      |                   | アクター名                       |
| 提供者     | 顧客管理システム                      |                   | アクター名                       |
| 入出力情報   | 受注顧客情報                        |                   | Dependency リソース             |

ビジネスプロセス分析では抽出できなかった“×”の項目が明確になる



# Dependencyモデル

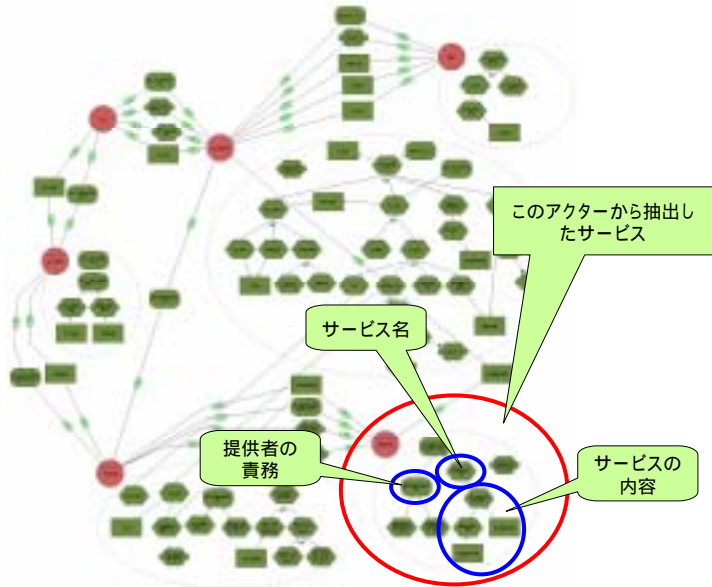




# Rationaleモデル



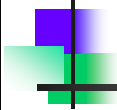
XML Consortium



© XML Consortium



XML Consortium



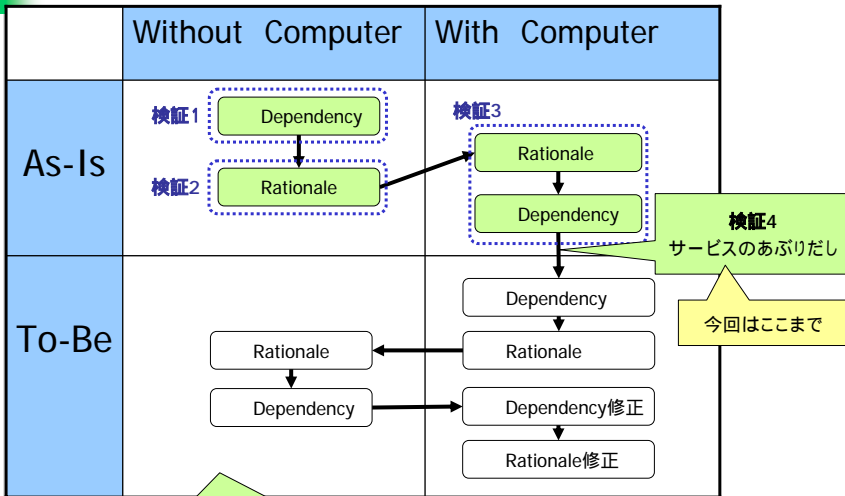
# 分析の流れ

© XML Consortium

# 分析の過程



XML Consortium



without Computerから書かなければいけないという手順は存在していない。  
経験則として、この順番が良さそうと言うだけである。

# Without/With Computer




XML Consortium

- without Computer
  - コンピュータを意識しないモデル
    - 業務の視点で作成することがポイント。
- with Computer
  - コンピュータシステムを意識したモデル
    - 目的(ゴール)を結びつけてアクターを1つにするのか別々にするのか判断できる。使わないシステムの開発を防止する効果がある。
    - 似ていても目的が異なるシステムは1つのアクターにはならない
- without Computer with Computer
  - システム化したい曖昧な要求を明確な要求に変える(ことが出来る)。
    - コンピュータシステムが導出されるものではない。
    - コンピュータシステムを業務視点で擬人化し、要求を明確化することが目的である。



## 検証結果



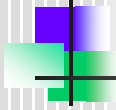
## 分析の対象

- シナリオの中で分析の対象とするシステム
  - 受注処理システム
  - 加工処理システム
  - 顧客管理システム

As-Isとして存在するシステムが各アクターとして、どんなサービスを提供しているのかを見る。

To-Beとして販売管理システムを新たに作ることを想定して既存システムとどう関係するのかを分析する。

© XML Consortium



# 検証1



- As-Is Dependency (without Computer)
  - 組織レベルの関係図からDependencyモデル (without Computer)を作成し、それぞれで何が表せるのかを整理する。
- 作成方法
  - 2チームに別れ同じ作業を行う
  - 作成の流れや作成結果の違いを考察する

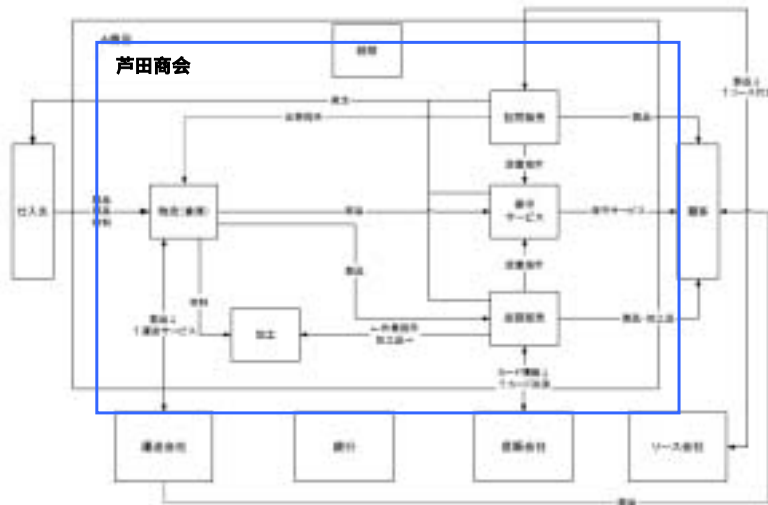


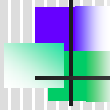
# 組織レベル(企業内)の関係図



一昨年前のモデリング検証タスクで作成したもの

(再掲)



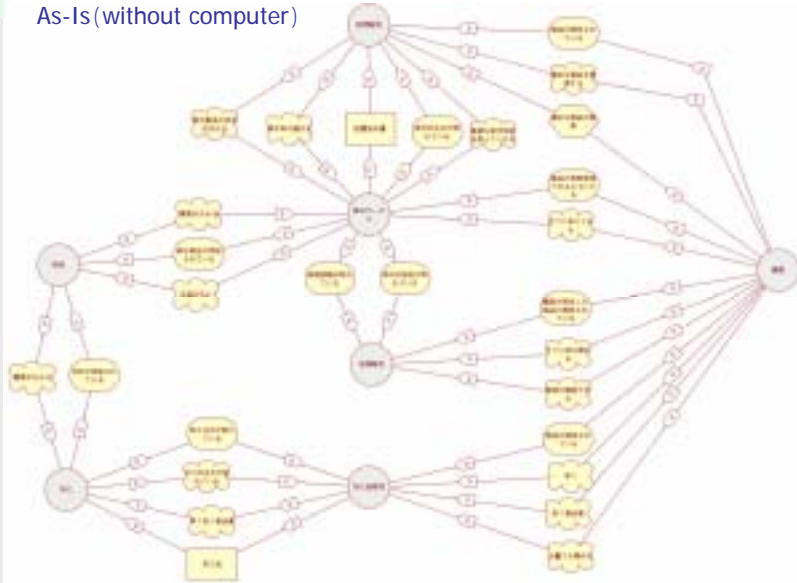


# Dependencyモデル(Aチーム)

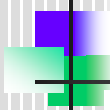


As-Is (without computer)

XML Consortium



© XML Consorti...

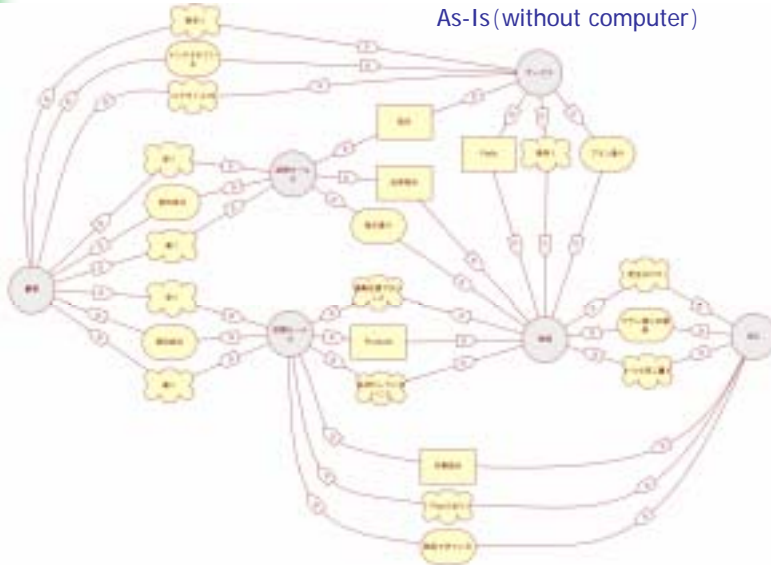


# Dependencyモデル(Bチーム)

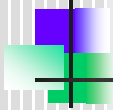


As-Is (without computer)

XML Consortium



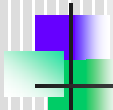
© XML Consor



## 検証1



- チーム別に作成してみたDependencyモデルについての考察
  - i\*法の欠点(使い方に工夫が必要そうな部分)
    - レイアウトのルールが無いため、同じ内容を表していても配置が異なると違って見えてしまうため、比較が難しい。
  - 2つのモデルの比較
    - 組織レベル関係図を基にしているため、アクターの決め方による差異はほとんどなかった。
  - Dependencyモデルの書き方のコツ
    - ゴールを始めに出す。
    - ソフトゴールは依存関係が逆方向になるものを出すようにする。
- 検証2はAチームのDependencyモデルを利用する。

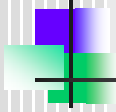


## 検証1



- Dependencyモデルを描いたときの思考
  1. 起点となるアクターを選択(顧客を選択)
  2. 起点となるアクターと関連のあるアクターを選択
  3. アクター間のゴールをモデリング
  4. アクター間のソフトゴールをモデリング
  5. アクター間のソフトゴールについて、4と依存関係が逆方向になるソフトゴールをモデリング

チーム間でのモデルの作成手順に大きな違いはなかった



## 検証1



- ゴール分析から組織の再編に波及するのか？
  - i\*法のモデリングはシステム開発の初期要求をモデリングするものであるため、組織体制などへ影響を及ぼすものではない。組織構成などある程度決まっている中でゴール分析を行う。
  - 事業の構造や組織間のゴールの設定は、経営企画段階で行われている。



## 検証2



- As-Is Rationale (without Computer)
  - Dependencyモデルの一部のRationaleモデルを作成
- 作成方法
  - 2チームに別れ同じ作業を行う
  - AチームのDependencyモデルを利用
  - 作成の流れや作成結果の違いを考察する

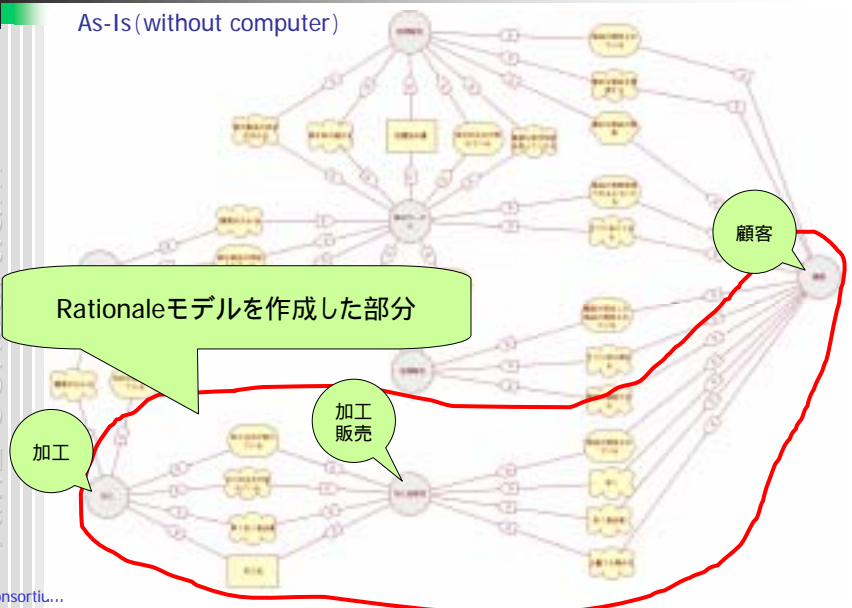


# Dependencyモデル(Aチーム)



As-Is (without computer)

XML Consortium



© XML Consorti...

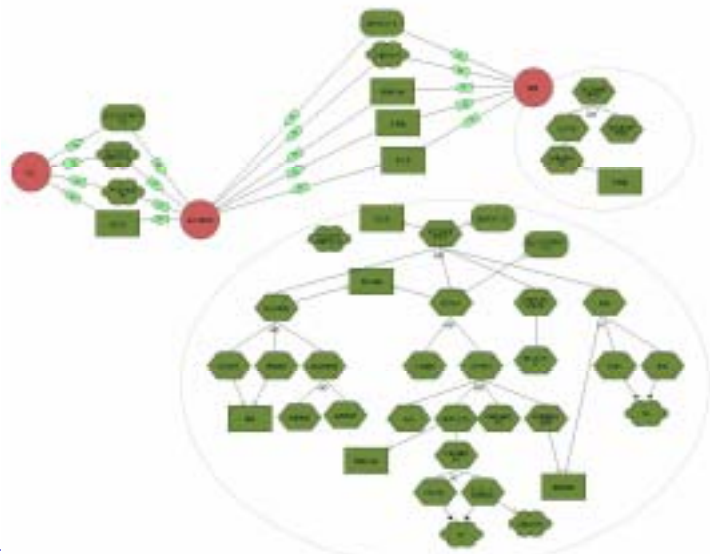


# Rationaleモデル

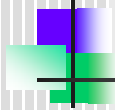


As-Is (without computer)

XML Consortium



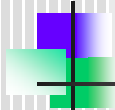
© XML Consortiur



## Rationaleモデル(Aチーム)



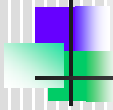
- 作成時の流れ
  1. アクター間のゴールからアクター内のトップタスクを設定
  2. タスクを分割
  3. アクター間のソフトゴールを見て抽出済みタスクとの関係を仮結線
  4. タスクと結線していないアクター間のソフトゴールを見てタスクを抽出(3と4の手順は確認時に誤りと分かり、これらの結線は全て削除した)
  5. 残りのタスク間の結線



## Rationaleモデル(Bチーム)



- 作成時の流れ
  - IT化を初めから意識してモデリングした。
  - モデリングの手順は両チームとも同じだった。

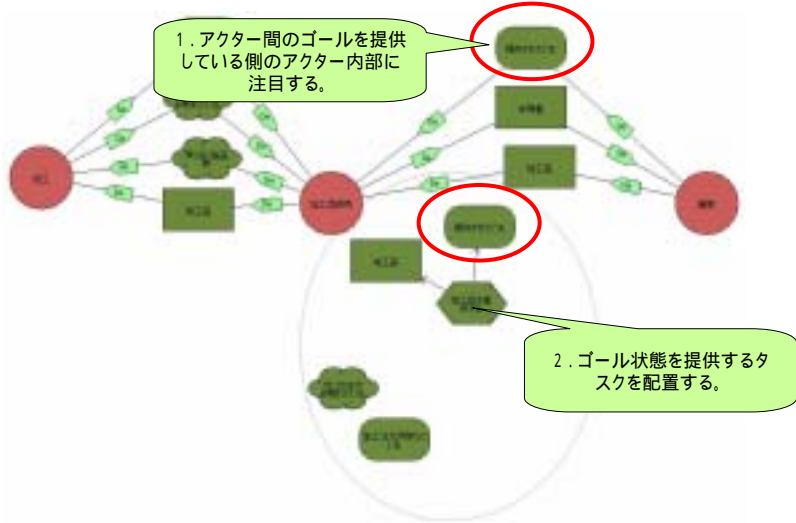


# Rationaleモデル



XML Consortium

## 作成手順1



# Rationaleモデル



XML Consortium

## 作成手順2



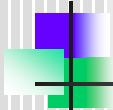
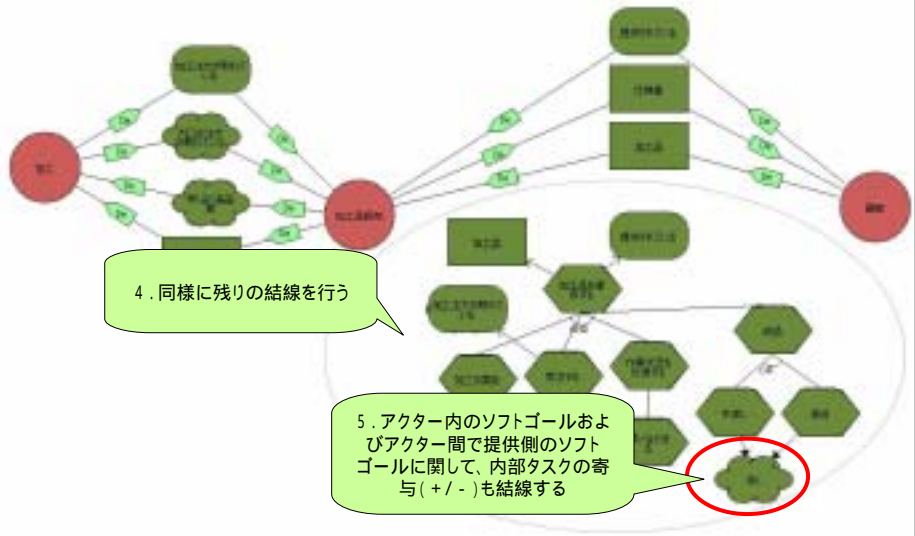




# Rationaleモデル



## 作成手順3



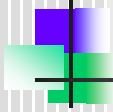
# Rationaleモデル



As-Is (without computer)

完成版 (再掲)  
2チームで作成したものを合成して1つのモデルにしたもの

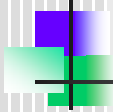




## Rationaleモデル(Aチーム)



- ポイント
  - +/-は、アクター内部のゴールに対してタスクがどう寄与するのかを表すために使う。アクター内部のタスクとアクター間のソフトゴールは通常結線しない。
  - タスク分割は、「アクター間のソフトゴールを達成する手段」という観点も考慮しながら導出していく。
  - ソフトゴールの挙げ方
    - 「達成できていないもの」を挙げていく。
  - "顧客"のようなシステムの外側のアクターの内部も「やってもらべきタスク」という観点で、タスクを抽出していく。



## Rationaleモデル(Bチーム)



- ポイント
  - タスク分割のポイント
    - 「分割した子タスクのみで親タスクの内容が実現できているか」を確認する。
  - タスク分割する過程で、孫タスクを描いていって、他のタスクとの関係からタスクをゴールに変えることもある(タスクよりもゴールとして表したほうが良い場合は変更する)。

## 検証3

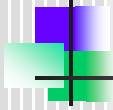


- As-Is RationaleおよびDependency (with Computer)
  - As-Is Rationale (without Computer)を元に作成する
- 作成方法
  - 1チームで作業を行う
  - 合成したRationaleモデルを利用
  - 作成の流れを考察する

## Rationale (with Computer)



- 作成時の流れ
  1. 現状のコンピュータシステムからアクターを追加
  2. Rationaleモデル (without Computer) から追加したアクター (加工管理、受注管理、顧客管理) 内のタスクを抽出
  3. 追加したアクターを含めたDependencyモデルを作成
  4. Rationaleモデル (without Computer) を更新

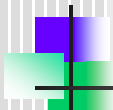
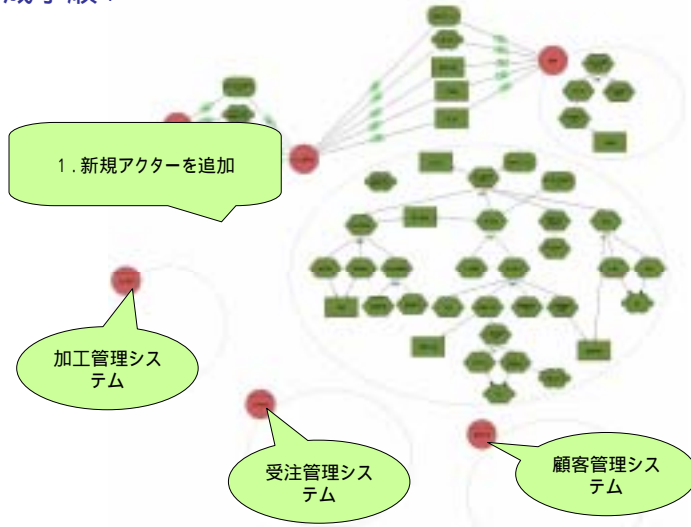


# With Computerの作成



XML Consortium

## 作成手順1

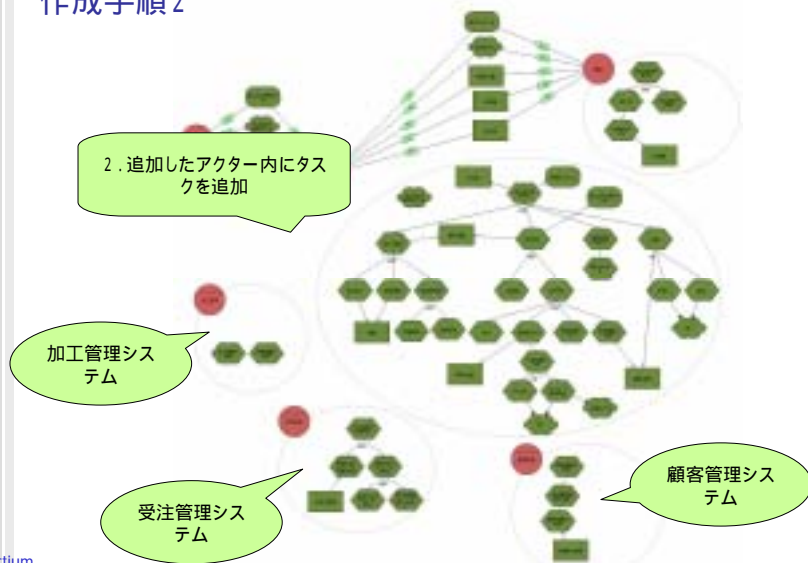


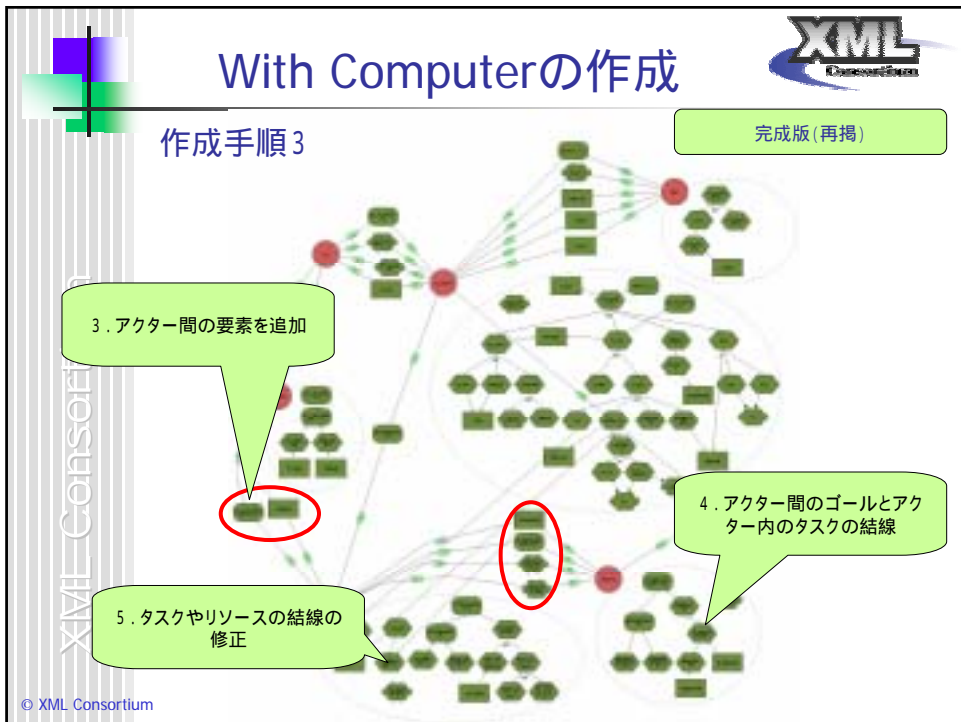
# With Computerの作成



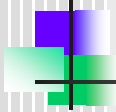
XML Consortium

## 作成手順2





- # Rationale (with Computer)
- ポイント
    - モデリングの仕方はwithout Computerと同じ。アクター間のゴールから始まって、ゴールを達成するタスクを洗い出す。
    - 目的外のシステムの仕様をタスクとして書き出すことはしない。
- © XML Consortium



## 検証4



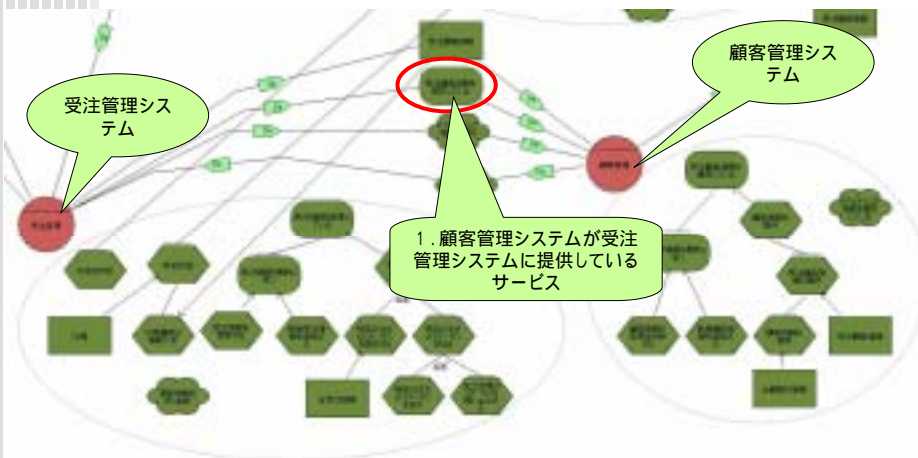
- サービスのあぶりだし手順
  1. Dependencyゴールとの関係から誰に提供するサービスか確定する
  2. ゴールを達成するためのタスクをサービス候補として抽出する
  3. 抽出したタスク(および子孫タスク)に関連するリソースからサービスの入出力情報を抽出する



## 検証4



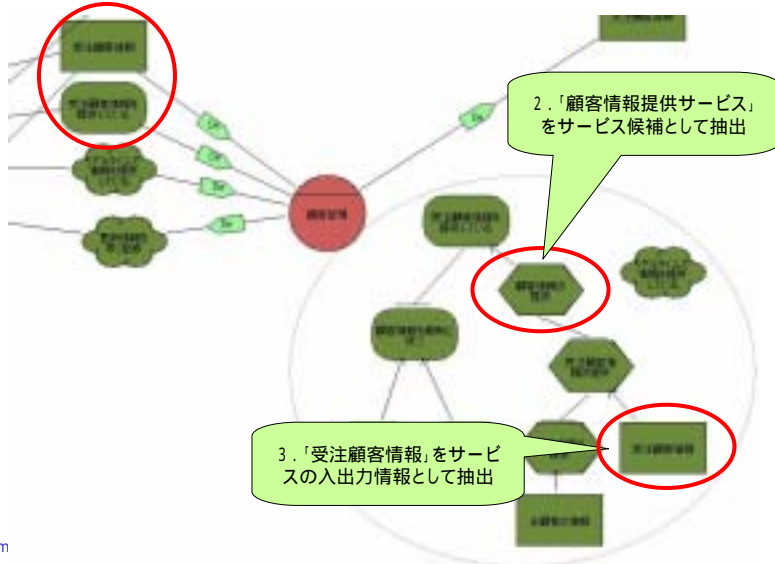
### 手順1



# 検証4



## 手順2



# 検証4



完成版(再掲)

| 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   | 顧客情報提供サービス                    |                   | Dependency ゴールと結線しているタスクの名称 |
| サービスの内容 | すべての顧客情報から受注顧客情報を検索して依頼者に提供する |                   | “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  | 注文書作成のための必要顧客情報を入手する          | ×                 | Dependency ゴールとリソース         |
| 提供者の責務  | 常に最新の顧客情報を提供する                | ×                 | 提供側の内部ゴール                   |
| 要望・制約   | 顧客情報をリアルタイムに提供する              | ×                 | Dependency ソフトゴール           |
| 利用者     | 受注管理システム                      |                   | アクター名                       |
| 提供者     | 顧客管理システム                      |                   | アクター名                       |
| 入出力情報   | 受注顧客情報                        |                   | Dependency リソース             |

## 検証5



| 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ると(同じ目的で利用するサービスは安定性が高いのではないか？という仮説)。利用者別のRationaleモデルを比較することであぶりだせないか期待している。                                |
| 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。  |

© XML Consortium

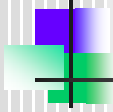


XML Consortium

## まとめ

© XML Consortium





## まとめ

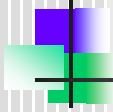


- i\*法の導入について
  - 「なぜ利用者はそのサービスを必要とするか」といった利用者側の視点が入ったことが大きな効果といえる。
- ビジネスプロセス分析を補完できるかという期待について
  - 現状は、i\*法によりAs-Isモデルを作成してサービスを抽出し、仮設の内容を紹介した状況。
  - 目的(ゴール)を中心に分析することで「本当に必要とされるものを提供する」「意味のないシステムを作らない」ことの一端を垣間見ることができた。
  - To-Beモデルの作成には至っていないため、ビジネスプロセス分析との比較については現状では結論は出せない。



## 今後の取り組み





## 今後の取り組み



- 分析の続き
  - i\*法による芦田商会モデルのTo-Beモデルの作成
  - 手順や注意点についての整理
- ビジネスプロセス分析による方法との比較、相関
  - 情報システムのプロセス設計やサービス設計のインプットとしてどのように役立つか