

# モデリング技術の 展開/動向/将来

2006年2月28日

株式会社豆蔵 取締役会長  
技術士(情報工学) 羽生田栄一

## 内容

- ◆起: 何のためのモデリングか?
- ◆承: 現在のモデリング技術
- ◆転: 今後の展望
  - アーキテクチャ
  - パターン
  - ビジネスモデリング
  - MDA
- ◆結: まとめ

# 起

何のためのモデリングか？

2006.2.28

Hanyuda Eiiti, Mamezou

3

## ソフトウェア開発に おける4+1つの複雑さ



## ソフトウェア工学と4つのP

- ◆ **People** (ピープル; ヒトと組織)
  - ◆ **Process** (プロセス; 作業と工程)
  - ◆ **Project** (プロジェクト; 計画・実行・評価)
  - ◆ **Product** (プロダクト; 製品と品質)
- ◆ 上記の時間的・空間的「複雑さ」をマネジメント(経営)する総合技術が“ソフトウェアエンジニアリング”である。
- ◆ 言語や技術だけではソフトウェアは作れない

5

Hanyuda Eiiti, Mamezou

2006.2.28



# 承

## 現在のモデリング技術

2006.2.28

Hanyuda Eiiti, Mamezou

6

## オブジェクト指向とモデリング1

- ◆ 業務の構成要素とソフトウェアのモジュールとができるだけ対応するようにシステムを設計
- ◆ 「対象領域に登場する重要な概念要素」=「明確なインターフェースを伴って内部状態を管理する安定したソフトウェア・モジュール」=オブジェクトの集団  
によってシステムの複雑さを抑え込もうという意図
- ◆ オブジェクト技術は、記号抽象化、手続き抽象化、データ抽象化、と進んできたソフトウェア概念の発展形として、従来の技術体系を継承して定義されている

7

Hanyuda Eiiti, Mamezou

2006.2.28



## 問題解決の方法

### 複雑な対象の管理の方法論

- ◆ **Divide** and Conquer 分割して解決せよ
- ◆ **Name** and Conquer 名前を付けて解決せよ
- ◆ **Abstract** and Conquer 抽象化して解決せよ
- ◆ **Exemplify** and Conquer 実例で検証せよ
- ◆ **Visualize** and Conquer 可視化して共働せよ

8

Hanyuda Eiiti, Mamezou

2006.2.28



めだつ



カンタン



# インパクト

11

Hanyuda Eiiti, Mamezou

2006.2.28



# 高橋メソッド

12

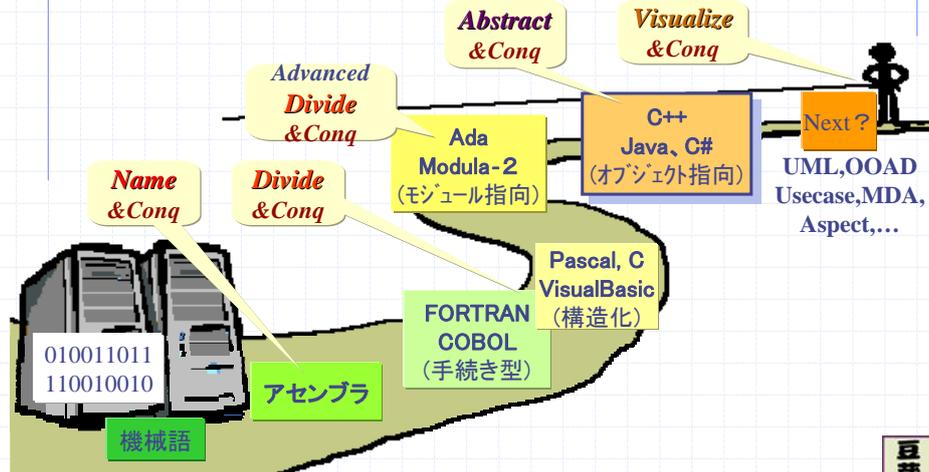
Hanyuda Eiiti, Mamezou

2006.2.28



## オブジェクト指向への道のり

- ◆「機械の動作」から「人間の思考」に近づく道程
- ◆Bit列→記号→手続き→モジュール→概念→

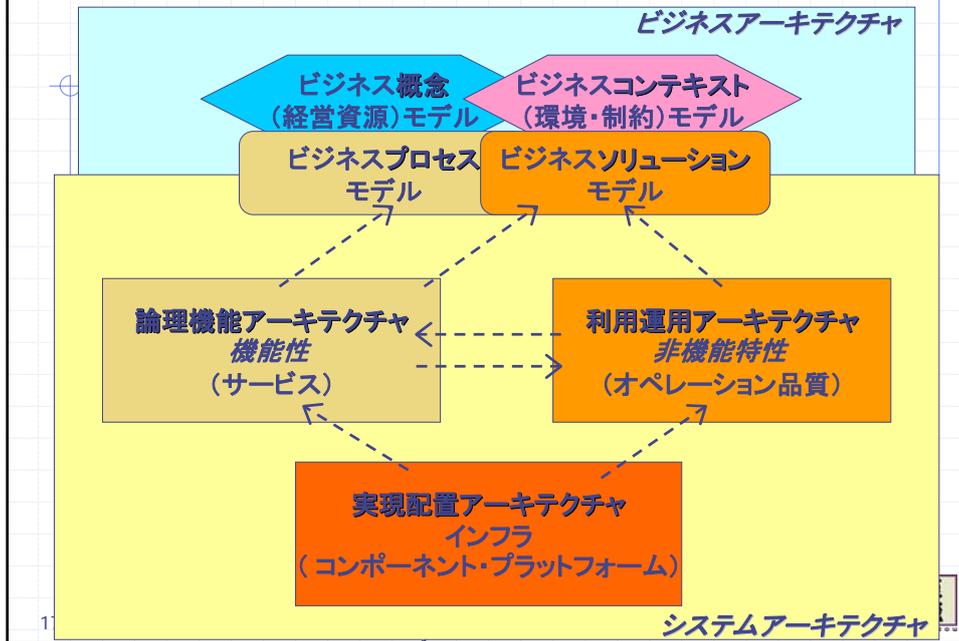


## オブジェクト指向とモデリング2

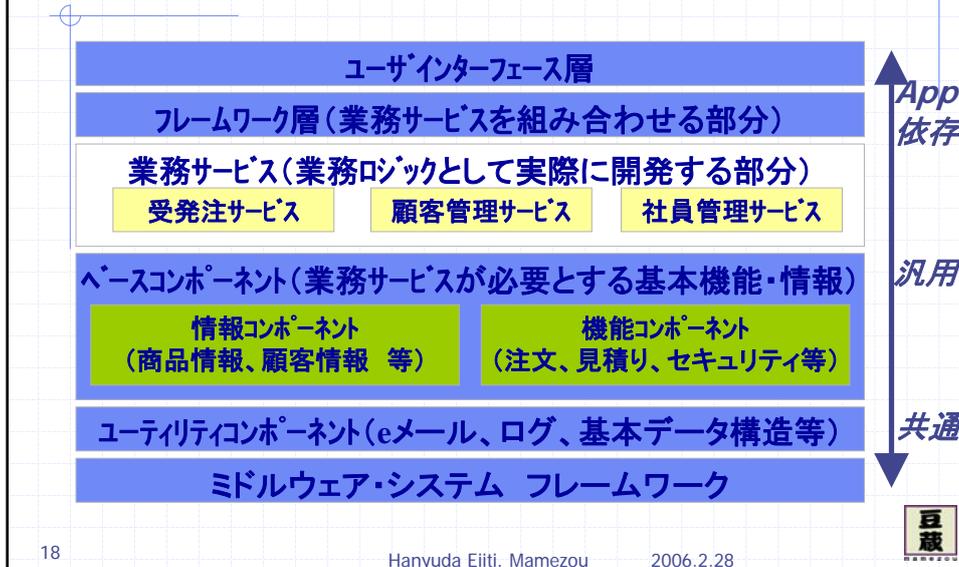
1. ユースケースによる要求のモデル化
  - ◆ 要求をユーザから見えるシステムの機能としてモデル
2. 業務のしくみと構成要素をシステムとモジュールに
  - ◆ システムを業務上の概念に対応するオブジェクトの集合で
3. UML: クラス図、オブジェクト図、コラボレーション図 etc
  - ◆ クラスに責務と役割を割り付けてコンポーネント化
4. 責務によるクラスの分類(ロール)とカプセル化
  - ◆ 他オブジェクトとの相互作用=関連により役割を演ずる
5. 汎化によるシステム構造の抽象化
  - ◆ 似たようなクラス同士を汎化関係により抽象化(抽象化手法)
6. ノウハウのパターン化、フレームワークとコンポーネント
  - ◆ アーキテクチャの枠組をパターン化し、大と小の部品化



## ビジネス/システムアーキテクチャの基本構成

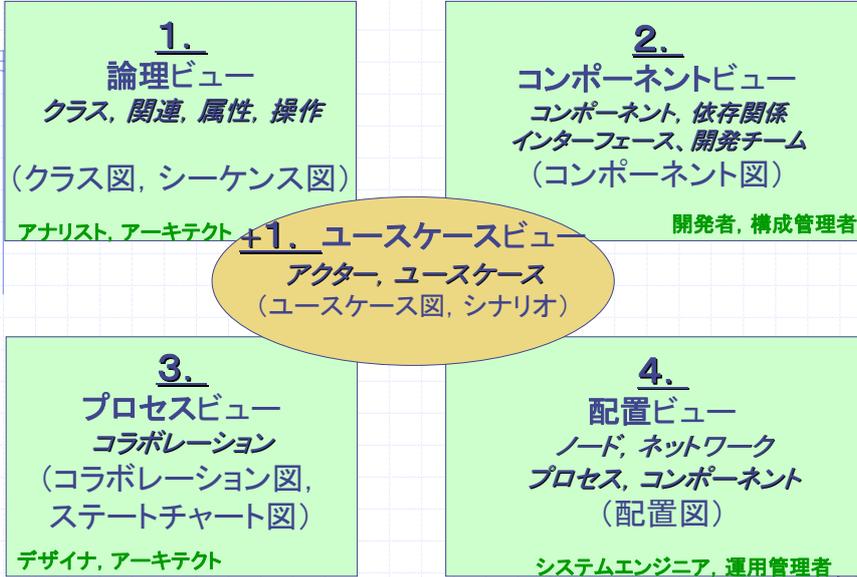


## アーキテクチャの基本構成例



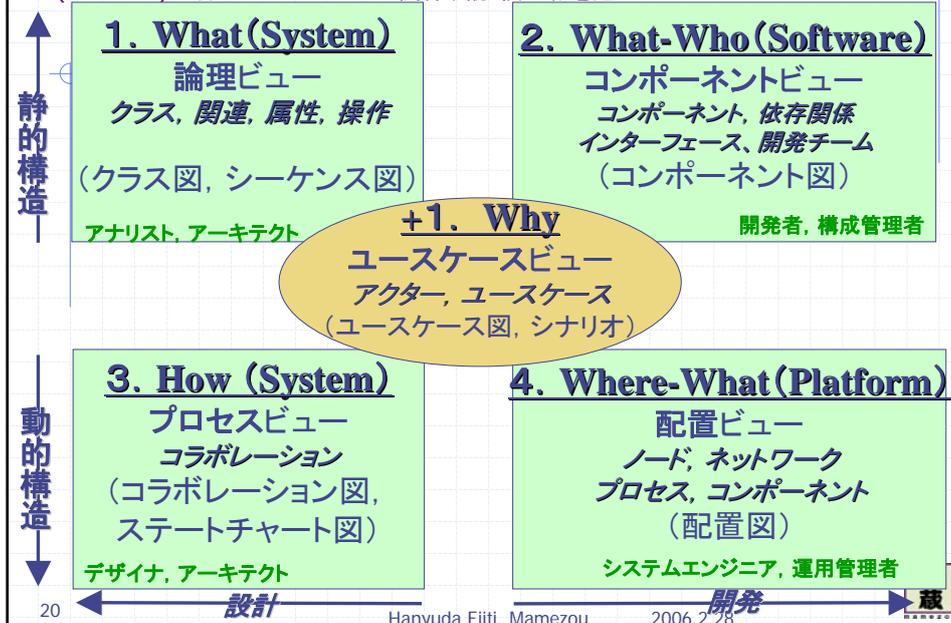
# ソフトウェアシステム4+1ビュー

◆ (注)オリジナル4+1ビュー



# ソフトウェアシステム4+1ビュー

(Kruchten)に羽生田が5W1Hとの関係、縦・横の軸を追加。



## ソフトウェアパターンの導入

- ◆ パターンを使えば、今までうまく伝えられなかった設計ノウハウ(の一部)を表現できる
- ◆ 組織内で、モデルやソースコードに乗せづらい非定型ナレッジを共有する仕組み
  - 暗黙知を形式知にしていく仕組み
- ◆ 知識のテンプレートかつ知識カタログ
  - 問題状況と解決戦術との対で記述
  - 既存のデザインノウハウのカタログ

21

Hanyuda Eiiti, Mamezou

2006.2.28



## パターンテンプレート

- 1) 名前と分類
- 2) 目的... そのパターンが何をするのか
- 3) 別名(もしあれば)
- 4) 動機... 問題を解くためのシナリオ
- 5) 適用可能性... どのような状況で使われうるのか
- 6) 構造... UMLなどを用いて、登場クラスの関係説明
- 7) 構成要素... 登場クラスの内容、責任について説明
- 8) 協調関係... 登場クラス間のインタラクション説明
- 9) 結果... 利点と欠点
- 10) 実装上の注意点
- 11) サンプルコード... C++, Java, Smalltalk
- 12) 使用例... 過去にどこにどのように利用されたか
- 13) 関連するパターン... 似ているが目的が違うもの等

22

Hanyuda Eiiti, Mamezou

2006.2.28



## ソフトウェアパターンの広がり

- ◆ ビジネスモデリングパターン
- ◆ アーキテクチャパターン
- ◆ アナリシスパターン
- ◆ デザインパターン
  - GoFデザインパターン
  - その他各種
- ◆ イディオム(=プログラミングパターン)
- ◆ プロセスパターン
- ◆ アンチパターン

23

Hanyuda Eiiti, Mamezou

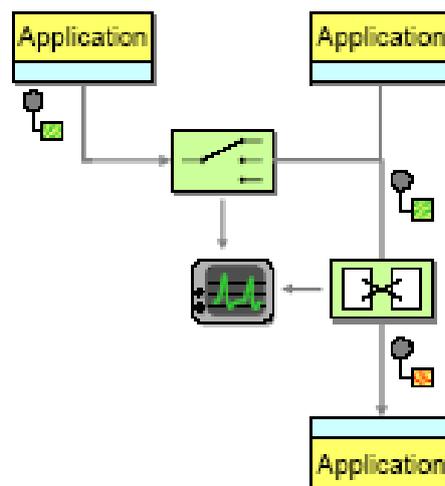
2006.2.28



## パターンの可能性: 非機能アспектへの表現

例: Enterprise Integrationパターン (Gregor Hohpe)

1. メッセージをつなげる
2. メッセージを設計する
3. メッセージを適切な相手先にルートする
4. 必要なフォーマットにメッセージを変換する
5. メッセージを生産し消費する
6. システムを調整しテストする



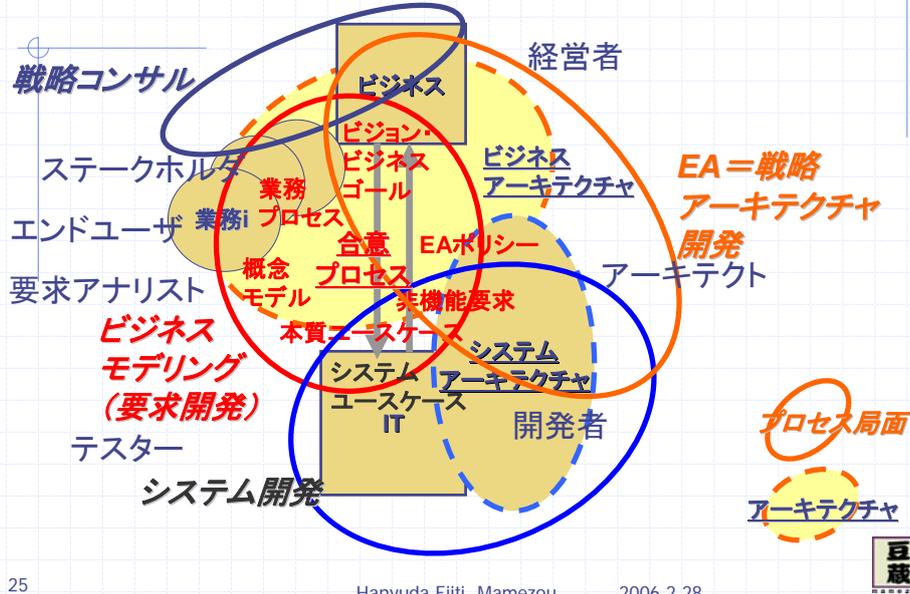
24

Hanyuda Eiiti, Mamezou

2006.2.28



# ビジネスとITの相互作用



25

Hanyuda Eiiti, Mamezou

2006.2.28

# ビジネスモデルの基本

◆基本は2つのモデル: 構造(静)vs. 振る舞い(動)

◆対象ビジネスに対する

## ■ ビジネス構造モデル

⇒クラス図

- ◆ そのビジネスで管理し利用される**基本概念(もの・こと)**
- ◆ 戦略的に重要な**ビジネスリソース**(ヒト, モノ, カネ, 情報, 技術, 組織)とそれらの間の**関係・制約**

## ■ ビジネス振る舞いモデル

⇒アクティビティ図(\*)

⇒相互作用図

- ◆ ビジネスを実行するために必要な**活動(ビジネスプロセス)**
- ◆ それらの**相互関係や順序・制約**, 入出力リソース

(\*)本来のオブジェクト指向であれば, 相互作用図のみでよいはずだが, ...  
ビジネス(業務)では, ワークフローの制御や改善が重要→アクティビティ図

26

Hanyuda Eiiti, Mamezou

2006.2.28

## 基本モデルを補う(1)

### ◆ゴール分析モデル

- 経営のビジョン⇔ビジネス戦略  
⇔ビジネスゴール⇔サブゴール
- 切り口には, BSC(バランススコアカード)が参考
  - ◆ 財務, 顧客, 内部プロセス, 人材と成長

### ◆ビジネスソリューションモデル

- ビジネスのための具体的なソリューション
  - ◆ 組織やサイト(場所)へのビジネスプロセスやビジネスリース(人・物・組織・カネ)の割り当て(配置; deployment)として表現するモデル
  - ◆ 組織(エージェント)に組織目標とその実現手段(プロセスやリソースなどのビジネスコンポーネント)を割り付ける
- サイト・エージェントを要素とする**コラボレーション図**
- サイト・エージェントにプロセス・リソースを割当てる**配置図**

27

Hanyuda Eiiti, Mamezou

2006.2.28



## 基本モデルを補う(2)

### ◆ビジネスシナリオ

- シナリオ思考法
  - ◆ 最終的には人間はストーリーで理解・判断する
- シナリオは企画と検証の基本
  - ◆ ビジネスの企画・設計⇒シナリオプランニング・シナリオ
  - ◆ システムの企画・設計⇒ユースケース
  - ◆ システム仕様化・検証⇒テストケース

### ◆ビジネスシナリオの検証

- シナリオの読み合わせ→ロールプレイ
- 質的シミュレーション→システムダイナミクス
- 概念プロトタイピング→ビジネスモデル構築キット

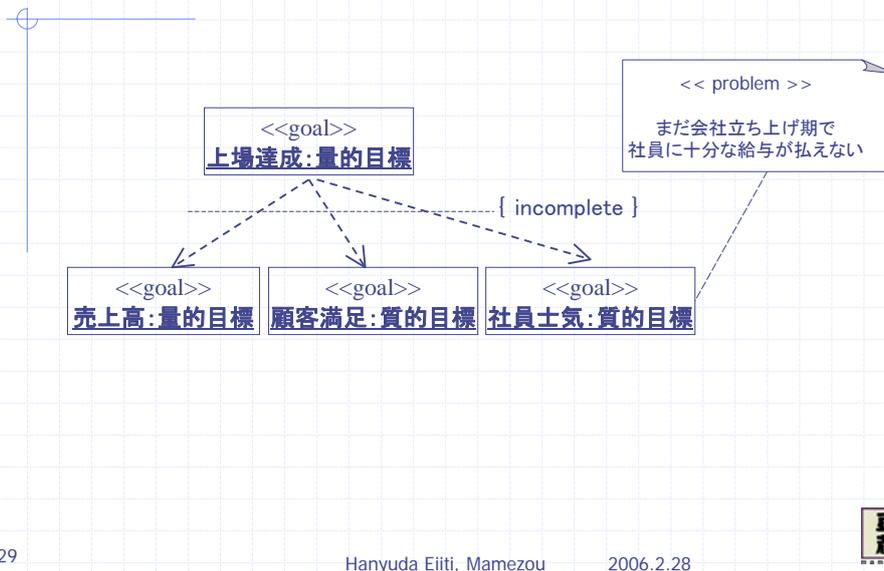
28

Hanyuda Eiiti, Mamezou

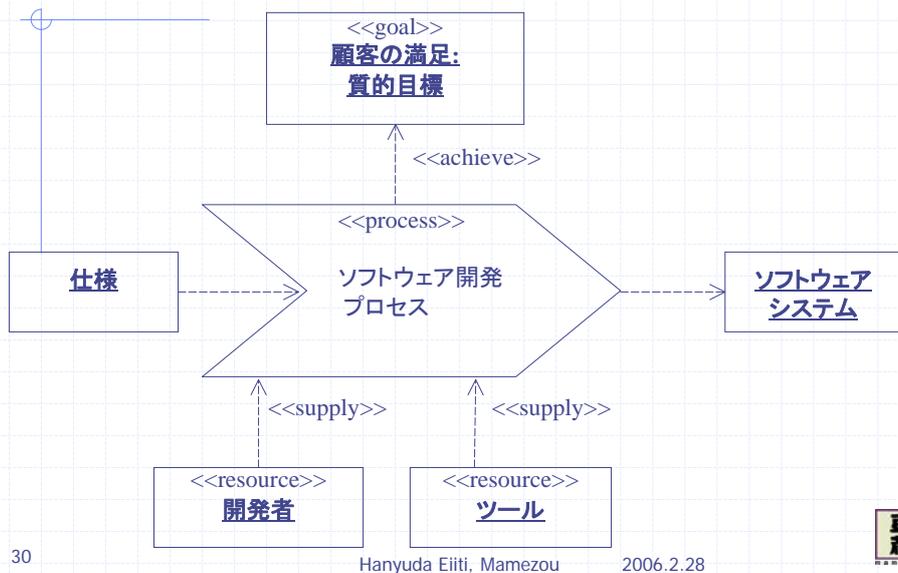
2006.2.28



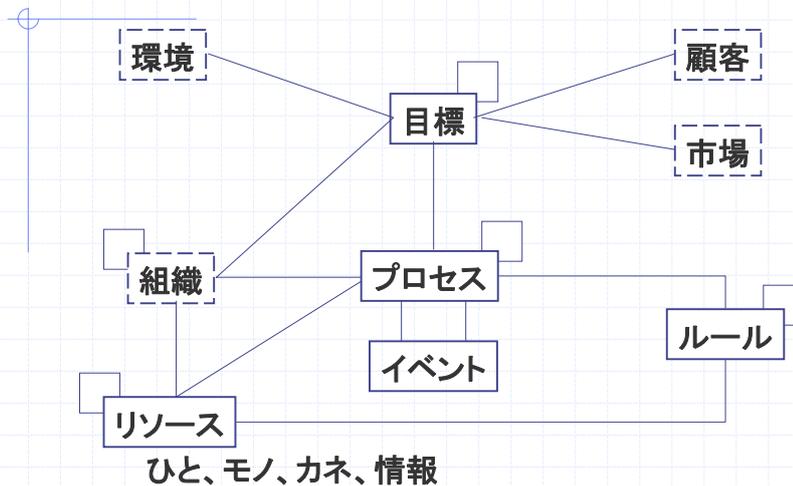
## ゴールのモデル化：ゴール/問題図



## ゴールのモデル化： ゴールをプロセスに設定した例



## ビジネスモデルの構成要素:メタモデル



31

Hanyuda Eiiti, Mamezou

2006.2.28



## UMLとビジネスモデル モデリングとWhy/What/How

### ◆ Why: ビジネス分析

- 機能: 業務フロー分析・ビジネス課題分析: アクティビティ図
- 構造: 業務ドメイン分析: 概念クラス図

### ◆ What: 要求分析

- 機能: システム要求分析: 要求を理解する: ユースケース図・仕様シーケンス図
- 構造: システム仕様分析: 設計を表現する土台: パッケージ図・仕様クラス図・仕様ステートチャート図

### ◆ How: 設計・実装

- 機能実現: 設計を理解する: 実現コラボレーション図
- 構造詳細: 実装を表現する土台: 実現クラス図

32

Hanyuda Eiiti, Mamezou

2006.2.28

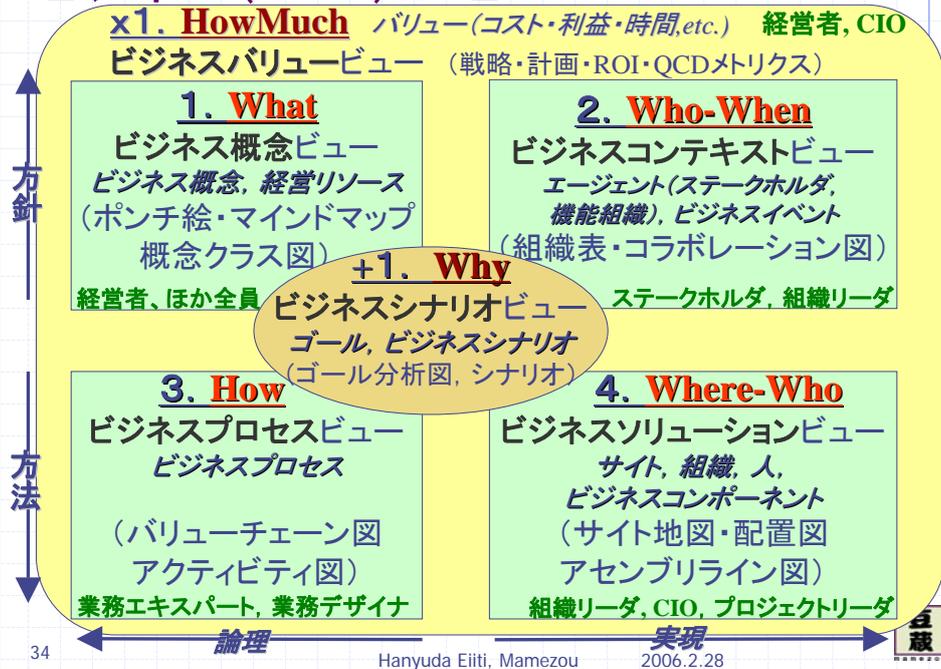


# ビジネスモデリングに対する複数の視点

- ◆ 4つのビュー: ビジネスアーキテクチャ基本4象限
  1. ビジネス概念: **What** ビジネス語彙, もの・こと分析, オントロジー
    - 概念構造
  2. ビジネスコンテキスト: **Who/When**
    - ◆ 環境-組織構造: ステークホルダー分析, 環境-企業-内部組織
  3. ビジネスプロセス: **How** 機能・価値フロー分析
    - ◆ 業務構造
  4. ビジネスソリューション: **Where/Who**
    - ◆ 実施構造: サイト・エージェント-ビジネスコンポーネント割付け
- ◆ +1のビュー ビジョン・ゴール・シナリオ
  - ビジネスゴール・シナリオ: **Why**
- ◆ x1のビュー 価値(利潤, コスト, 時間, 質)
  - ビジネスバリュー: **HowMuch**



# ビジネス(4+1)x1ビュー



# ビジネスモデリングで利用するUMLダイアグラムの例

ビジネスモデル名称	UMLダイアグラム	用途
ビジネス概念図	クラス図	ビジネス上の重要な概念やリソースとそれらの間の意味的な関係を表現
ビジネスコンテキスト図	コラボレーション図	顧客を含むステークホルダーや組織の間のビジネス上のやりとり(ビジネスイベント)による協調を表現：外部コンテキストと内部コンテキスト
ビジネスプロセス図	アクティビティ図	業務のワークフローとその中の各ビジネスプロセスの実行主体(スイムレーン)を表現
ビジネス配置図	配置(deployment)図	ビジネスを実施する上でのアクターやビジネスコンポーネントの組織への配置，組織のサイトへの配置を表現
ビジネスユースケース図	ユースケース図	企業や組織をシステムとみなしたときの外部アクター(ステークホルダー)と彼らにその企業や組織が提供すべきビジネス上の価値としてのビジネスユースケースを表現
アセンブリライン図	アクティビティ図とパッケージ図の拡張	プロセスとそこで利用され生成されるビジネスリソースや機能組織との関係を表現

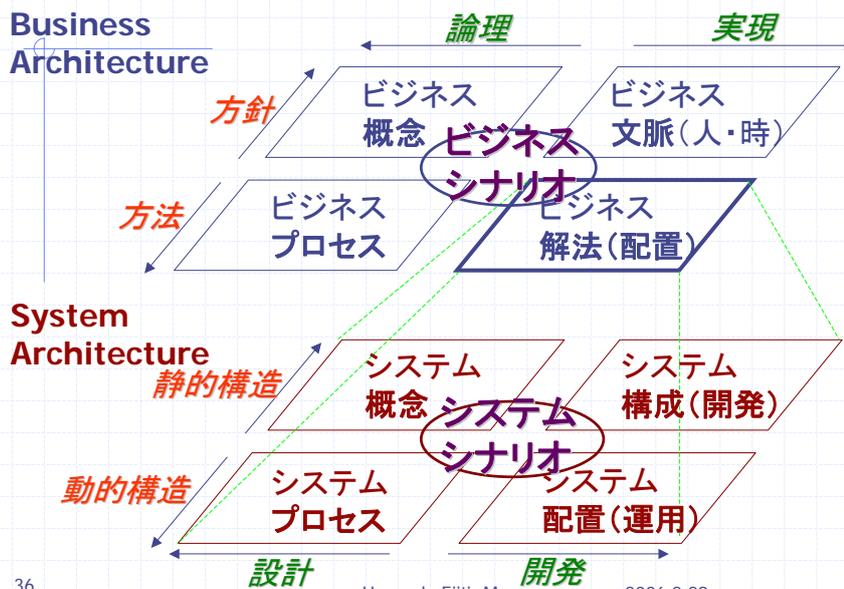
35

Hanyuda Eiiti, Mamezou

2006.2.28



# ビジネスBAとシステムSAの対応関係



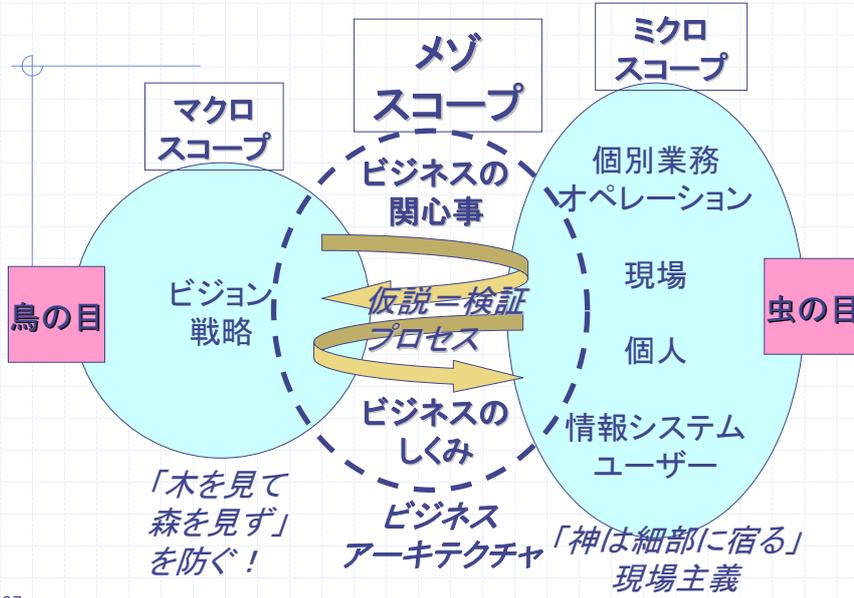
36

Hanyuda Eiiti, Mamezou

2006.2.28



### 3つのスコープ: マクロ・メゾ・ミクロ



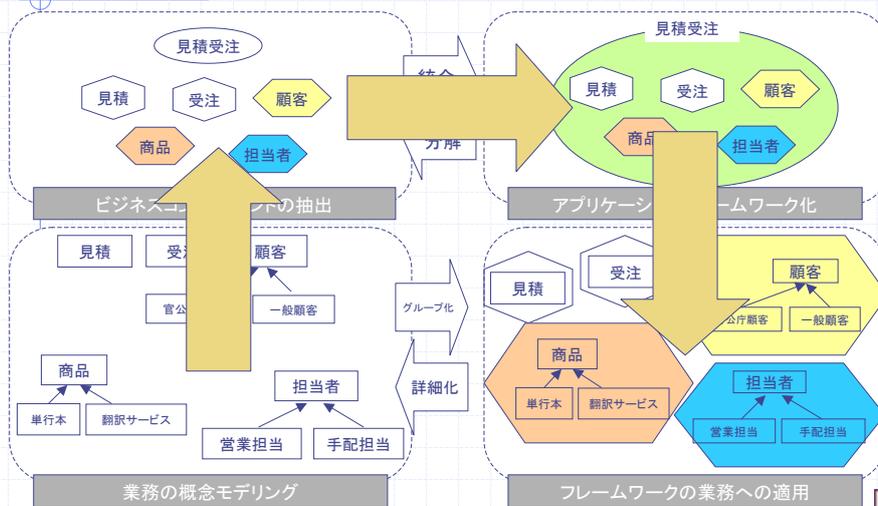
37

Hanyuda Eiiti, Mamezou

2006.2.28



### 業務モデルとコンポーネント



38

Hanyuda Eiiti, Mamezou

2006.2.28



## MDAとは

- ◆ モデルを中心にシステム開発を行うためのシステムアーキテクチャを標準化するOMGイニシアチブ
- ◆ 3つのモデル抽象(ビューポイント)レベル
  - CIM (Computation Independent Model)
    - ◆ ビジネスプロセス、ビジネスユースケースなどコンピュータに依存しない業務モデル
  - PIM (Platform Independent Model)
    - ◆ プラットフォーム (CPU、OS、ミドルウェア、言語) に依存しないモデル
    - ◆ 実装技術に非依存。従来開発での分析モデルに相当。
  - PSM (Platform Specific Model)
    - ◆ プラットフォームに依存したモデル
    - ◆ 設計判断の記述。従来の開発でいう設計モデルに相当。

39

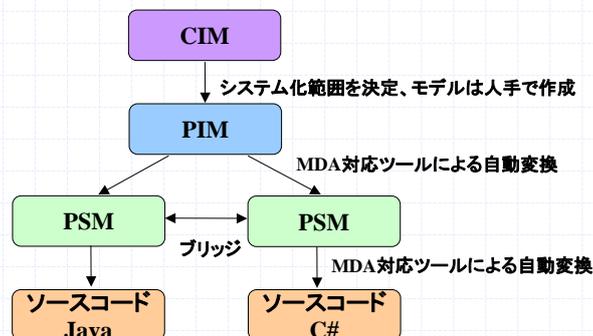
Hanyuda Eiiti, Mamezou

2006.2.28



## MDAにおけるモデル変換

- ◆ モデル間の関係



40

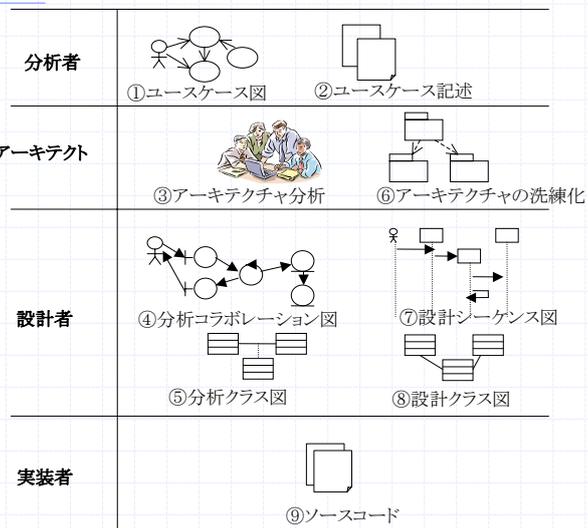
Hanyuda Eiiti, Mamezou

2006.2.28



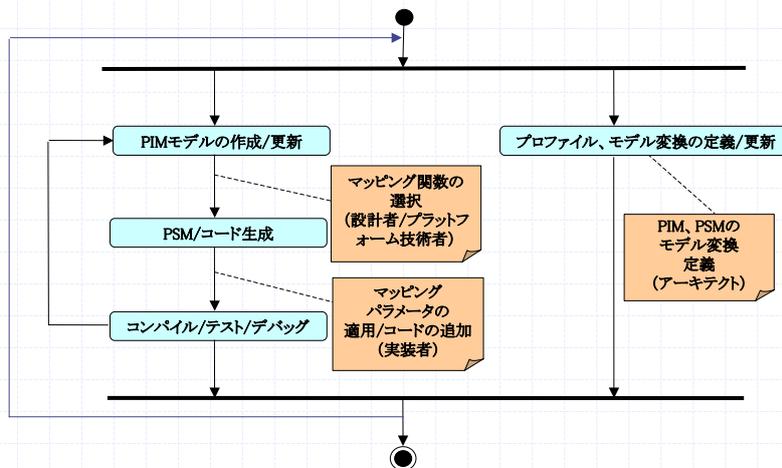
# 人手による開発のケース

## ◆RUPによる開発プロセスの概要



# MDAによる開発のケース

## ◆MDAによる開発プロセス



# 糸口

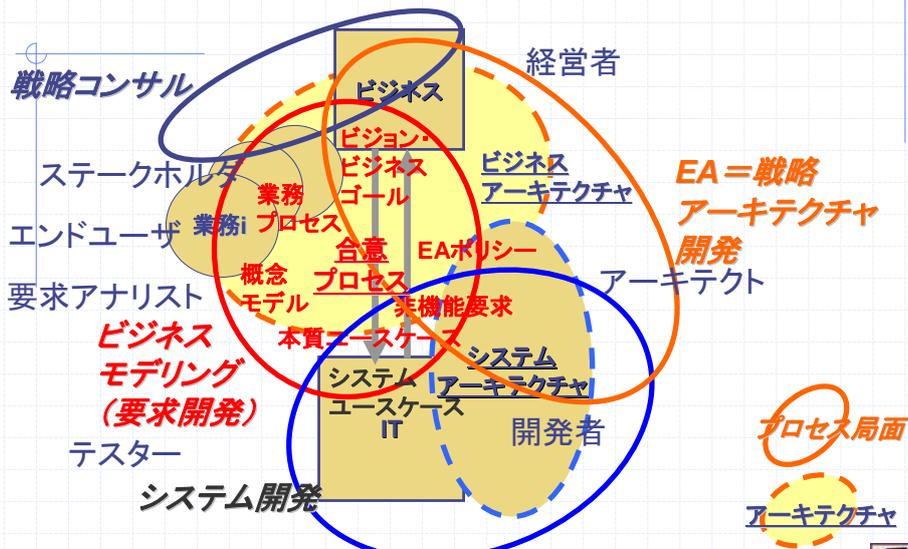
原題:ストーリーパターン(4)

2006.2.28

Hanyuda Eiiti, Mamezou

43

## ビジネスとITをとりまく生態系



44

Hanyuda Eiiti, Mamezou

2006.2.28



# ソフトウェアと合意形成プロセス

- ◆ソフトウェア開発(企画, 要求, 仕様, 設計, 実装, リリース, 保守, 拡張)は様々な利害関係者(stakeholders)
- ◆ソフトウェアは目に見えない
- ◆さまざまな非機能要求

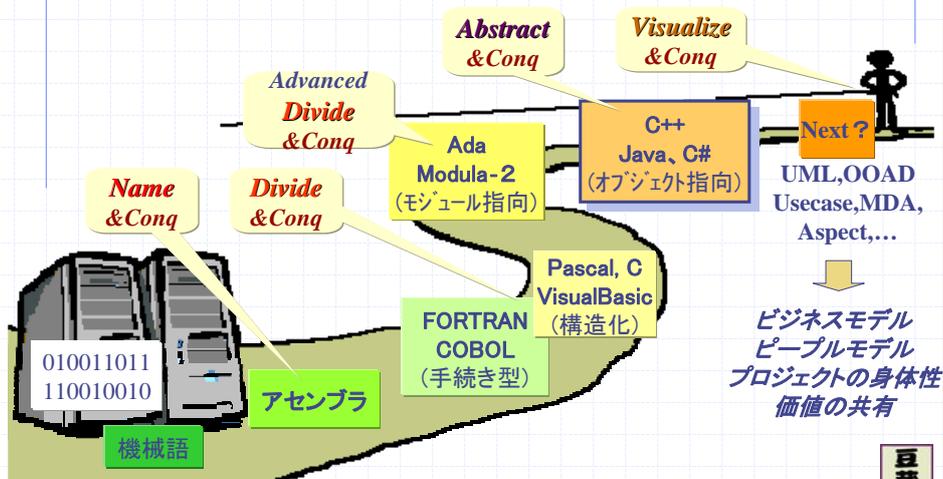
## 要求とアーキテクチャの適切かつ迅速なバランス

- ◆ユーザー参加型の要求ワークショップのファシリテータ
  - 自己組織型の合意形成プロセス推進の必要性
- ◆経営とITのリアルタイム同期というニーズ
  - 要求と設計をつなぐ技術通訳=アーキテクトの必要性

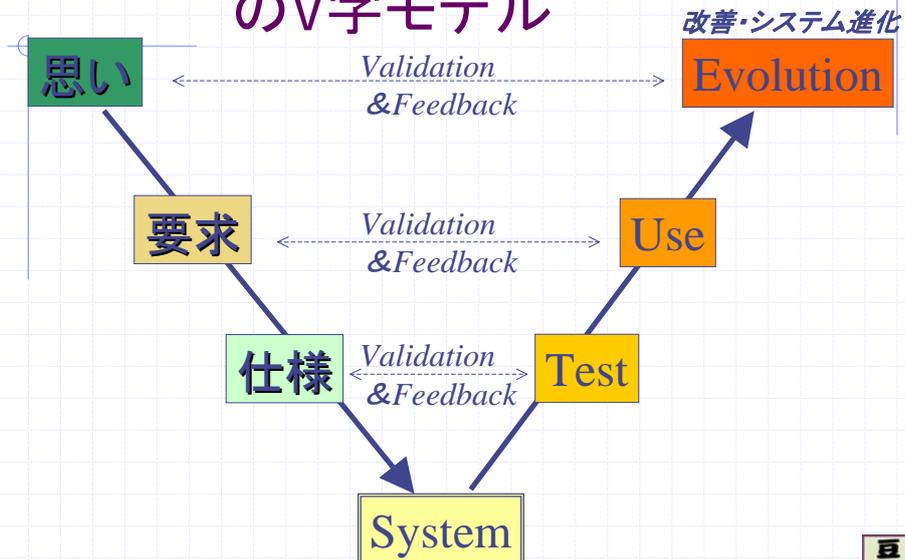


# 複雑さの管理対象の上昇プロセス

- ◆「機械の動作」から「人間の思考」に近づく道程
- ◆Bit→記号→手続き→モジュール→概念→社会



## まとめ：思い・要求とシステム進化のV字モデル



47

Hanyuda Eiiti, Mamezou

2006.2.28



## 参考文献

- ◆ 羽生田栄一ほか：『ソフトウェアの匠』，日経BP社，2004
- ◆ Eric J. Braude：ソフトウェアエンジニアリング 実践的オブジェクト指向技術体系，翔泳社，2004
- ◆ Wycoff：マインドマッピング-創造性を全開する脳力活用法，日本教文社，1985
- ◆ 戸田保一・飯島淳一編：ビジネスプロセスモデリング，日科技連，2000
- ◆ Eriksson/Penker：UMLによるビジネスモデリング，ソフトバンク，2002
- ◆ 中埜博：パタン・ランゲージによる住まいづくり，井上書院，1988

48

Hanyuda Eiiti, Mamezou

2006.2.28

