

オブジェクトモデル分科会

オブジェクトモデリング分科会 のアプローチ

2001年10月3日

児玉公信

(株)エヌ・ケー・エクサ

目次

1. なぜオブジェクトモデルなのか
2. オブジェクトモデリングの方法
3. 何をモデル化するか
4. ロードマップ

1. なぜオブジェクトモデルなのか

1.1 モデリング

1.2 インタフェース

1.3 XMLとインタフェース

1.1 モデリング (1)

□モデルとは

□ある人にとっての、ある状況あるいは状況の概念の明示的な解釈 (Wilson, B., 根来監訳:「システム仕様の分析学」, 共立出版, 1996)

□システムの鳥瞰図

□モデルの分類

□アイコン型モデル:ミニチュア模型 (or拡大模型)

□類比型モデル:実物の代わり

□分析型モデル:数式表現

□概念モデル:

1.1 モデリング (2)

□ 概念モデル

- 関心領域についての考察を助ける
- 概念を図式化する
- 構造と論理を定義する
- 設計の前提とする

分析, 設計, 実装
の連続性

□ メンタルモデル

- 思考操作のための概念構造の表現
 - 捨象, 単純化
 - 構造の理解 → シミュレーション, 予測

□ 認識の共有

- モデリング言語

1.1 モデリング (3)

□モデルの認識主体

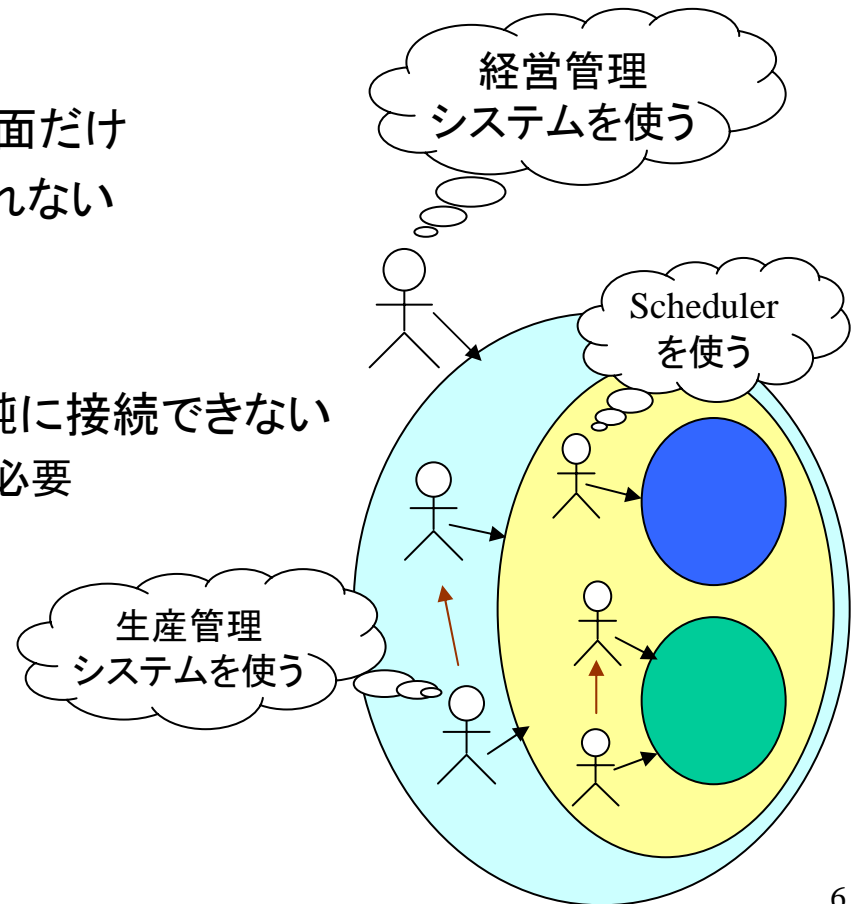
□システムのOwner/Observerの視点

□操作対象

- 見ることのできるのは境界面だけ
- 認識主体はモデルには現れない

□アクタと目的

- 階層性
- 視点が異なるモデルは単純に接続できない
 - 意味のギャップを埋める必要



1.2 インタフェース (1)

□境界のあるところにインタフェースがある(接面)

□なぜ境界を設けるのか

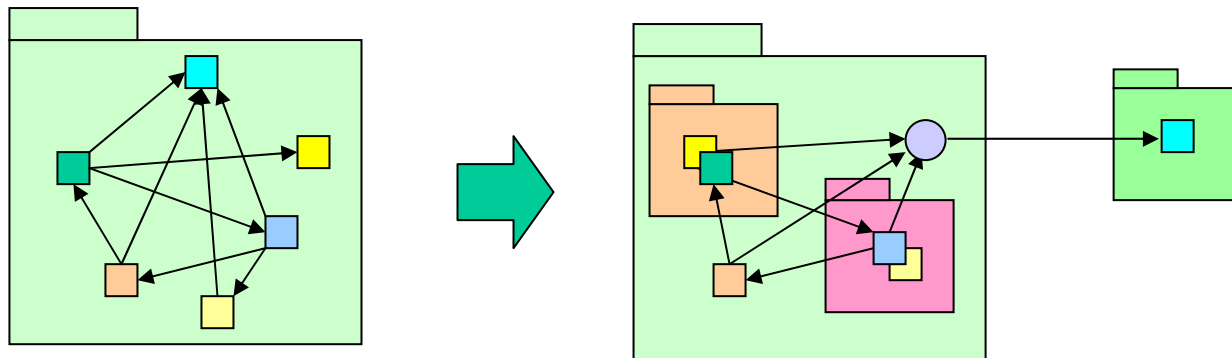
□複雑さのコントロール

□コミュニケーション量の最小化(coupling)

□機能の凝集化(cohesion)

□オーバーヘッドの減少(機能のサイズ)

□できるだけ適切な境界を探す



1.2 インタフェース (2)

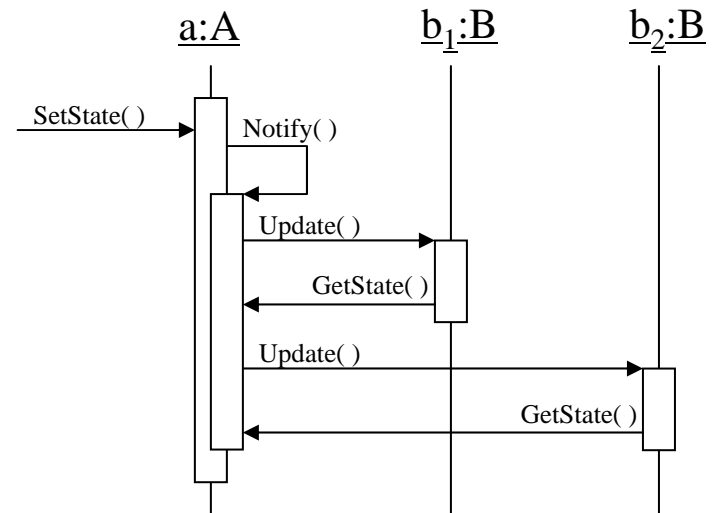
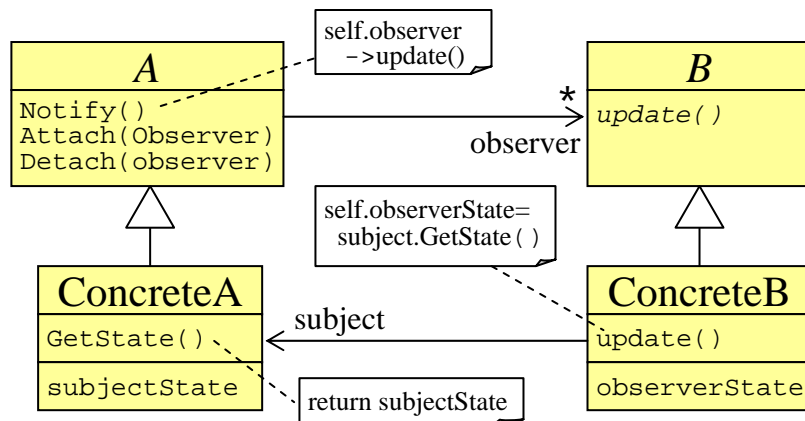
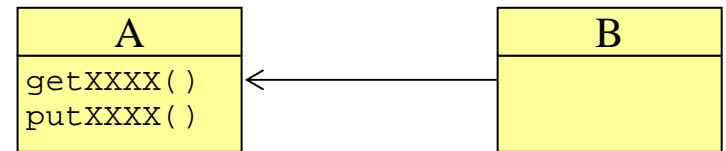
□境界のつなぎ方

□一方向の可視性

□read only

□Observer pattern (MVCモデル)

□モジュール間の相互作用



1.2 インタフェース (3)

□ データの見せ方

□ Application Façade

□ モジュールが持っている内部データ構造ではなくて、どう見せるか

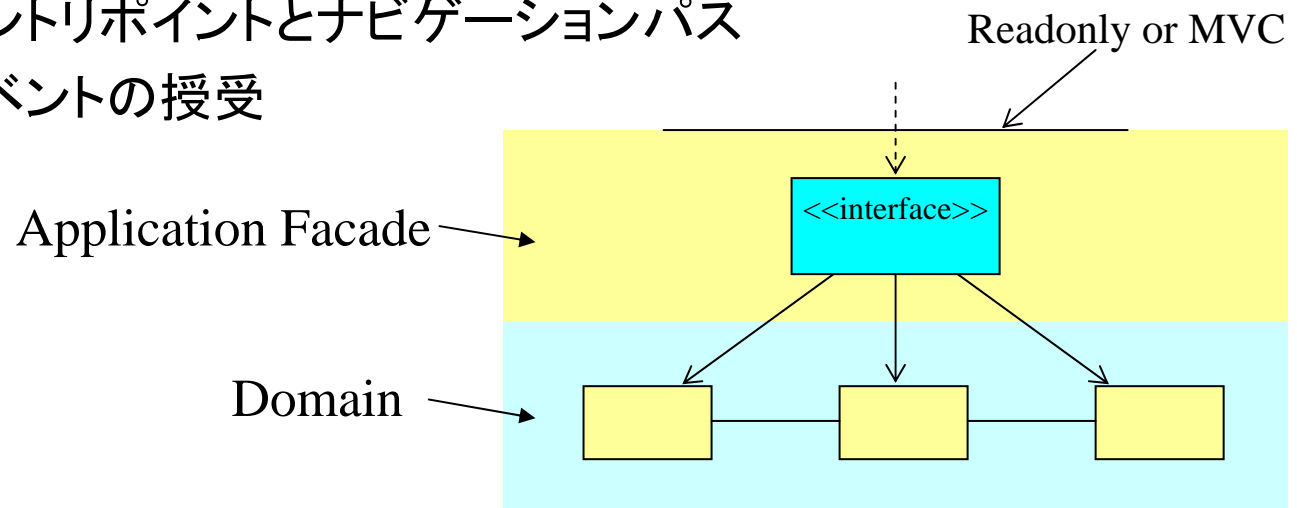
□ 効率や実装上の都合

□ プロトコル

□ 全体のデータ構造を一度に渡すのは困難

□ エントリーポイントとナビゲーションパス

□ イベントの授受



1.3 XMLとインタフェース

- インタフェースの実装方法のひとつ
 - Application Façadeの静的側面の写像
 - 動的側面については何らかのプロトコル定義
- XMLとインターネットコア技術分科会との調整必要

2. オブジェクトモデリングの方法

2.1 方法論

2.2 静的側面

2.3 動的側面

2.4 静的側面と動的側面の整合

2.1 方法論 (1)

- モデリングのための本質的な方法論はない
 - どのような概念を見出すか
 - どのようなビジネスプロセスを見出すか
 - 世界観 (Weltanschauung) と密接に関連
- モデリング手法ならあることはあるが
 - 名詞抽出法
 - CRCカード
 - リファクタリング
 - ユースケース
 - ただし、ユースケースだけではモデリングはできない

2.1 方法論 (2)

□ ユースケース

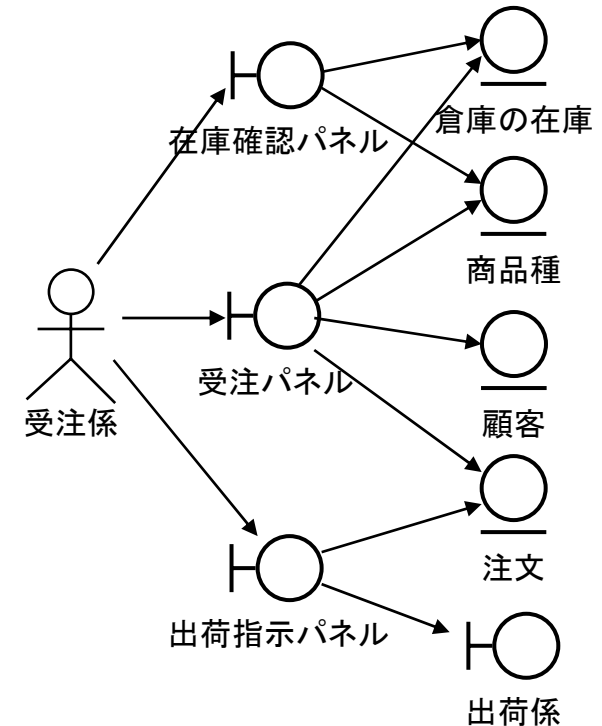
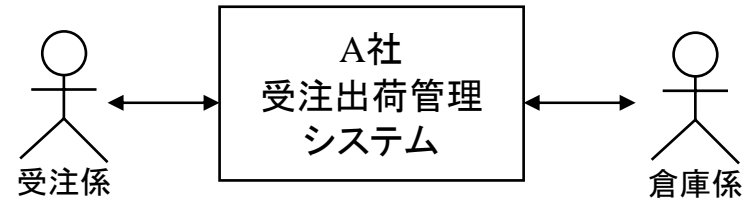
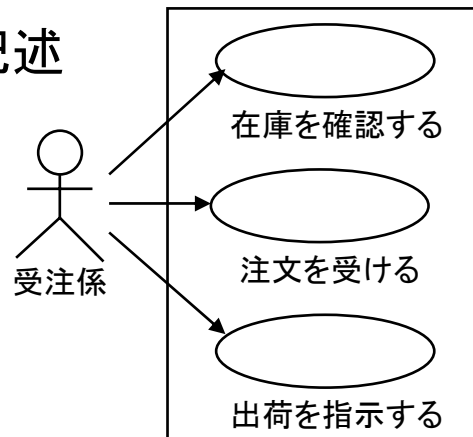
□ ユースケースのレベル

- 概観図
- 要求レベル
- 分析レベル

□ 要求レベル

□ 要求の単位

- ユースケース記述
- 要求の境界

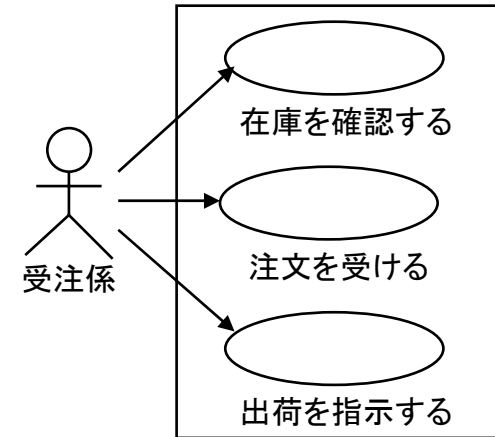


2.1 方法論 (3)

□ ユースケース

□ モデルの動的側面の記述の一部

□ モデルの静的側面を検証, 補強するものとして使う



ユースケース名
ユースケースの目的
ユースケースのアクタ
変換内容
 事前条件 (pre-condition)
 主系列 (main sequence)
 事後条件 (post-condition)
 代替系列 (alternative sequence)
備考

2.2 静的側面 (1)

- 概念的観点
 - データ保有の側面
 - 静的側面
 - 概念と概念間の構造
 - entity-set (Chen, 1976)
 - クラスとして実装されうる / テーブルとして実装されうる
- UMLのクラス図の記法を使う
 - 「型」であることを明示することになっているが...
 - <<type>>ステレオタイプ
 - 型図

2.2 静的側面 (2)

□ 型

□ 概念

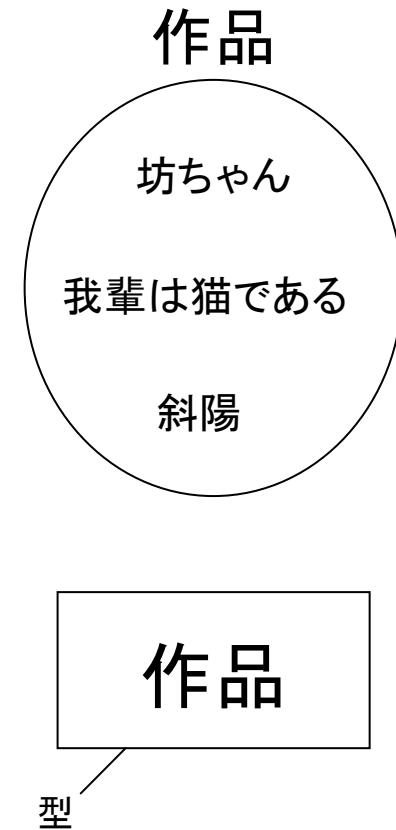
□ 集合 *cf.* Entity-set (Chen, 1976)

□ 集合を識別する属性

□ 正規形

□ インスタンス

□ 集合の要素



2.2 静的側面 (3)

□ 関連

□ リンク

□ インスタンスの対応関係

□ *source* → *target*

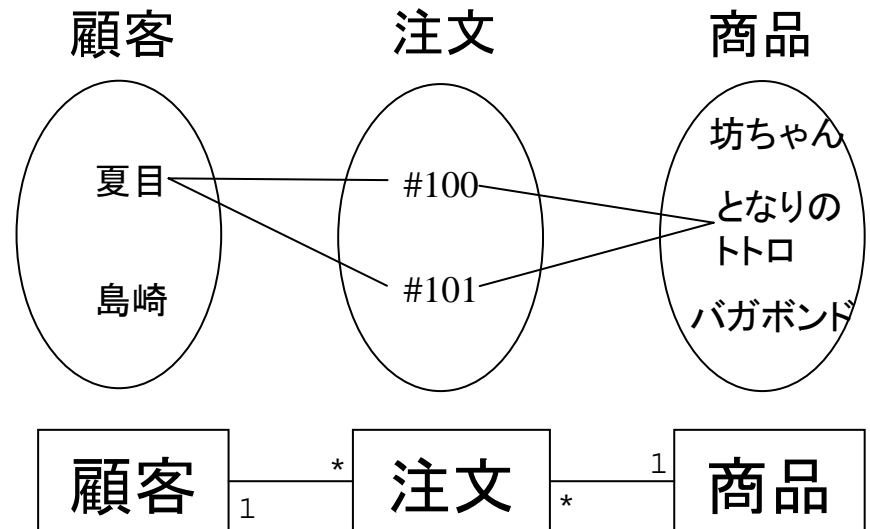
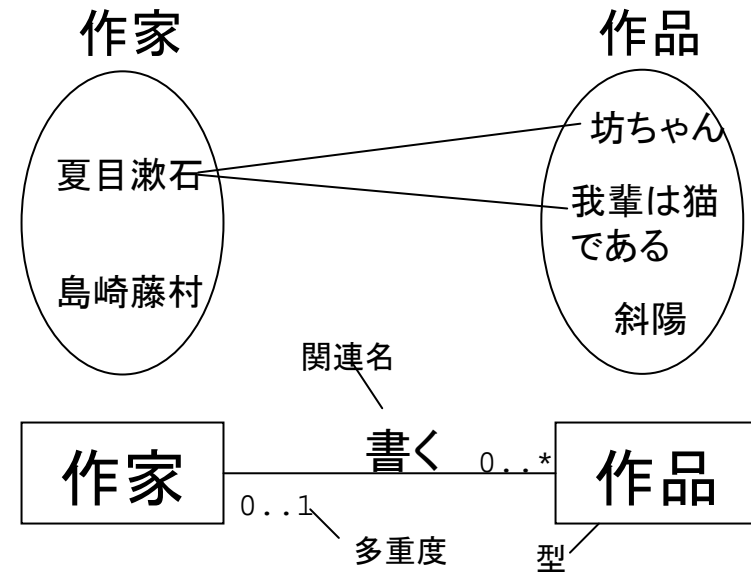
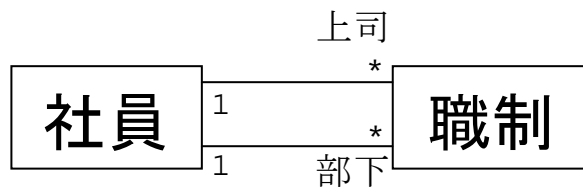
□ 関連

□ リンクの集合

□ 多重度

□ ロール

□ *s* にとっての *t* の役割



2.2 静的側面 (4)

□ 関連

□ 集約

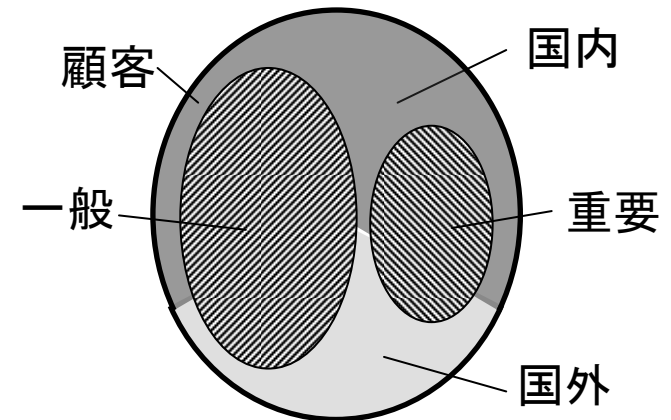
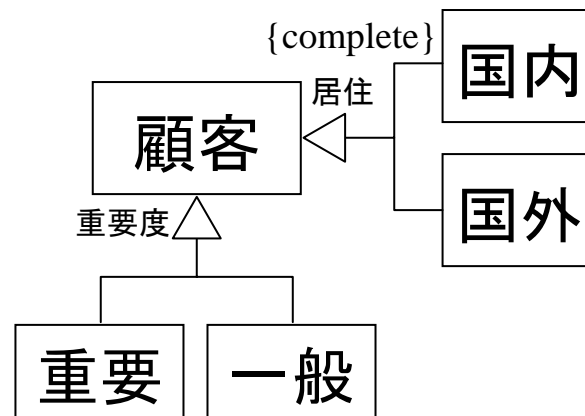
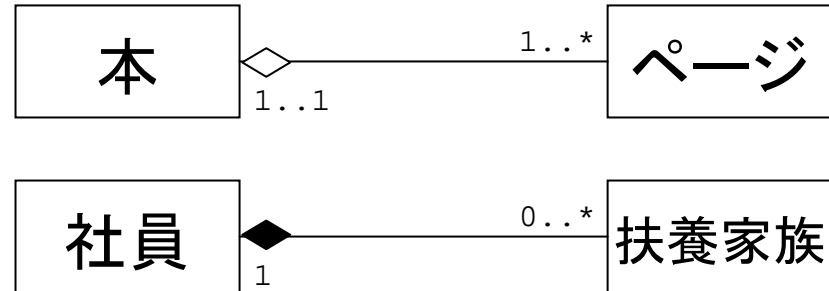
□ サブタイプ

□ ステレオタイプ

□ 制約とルール

□ OCL

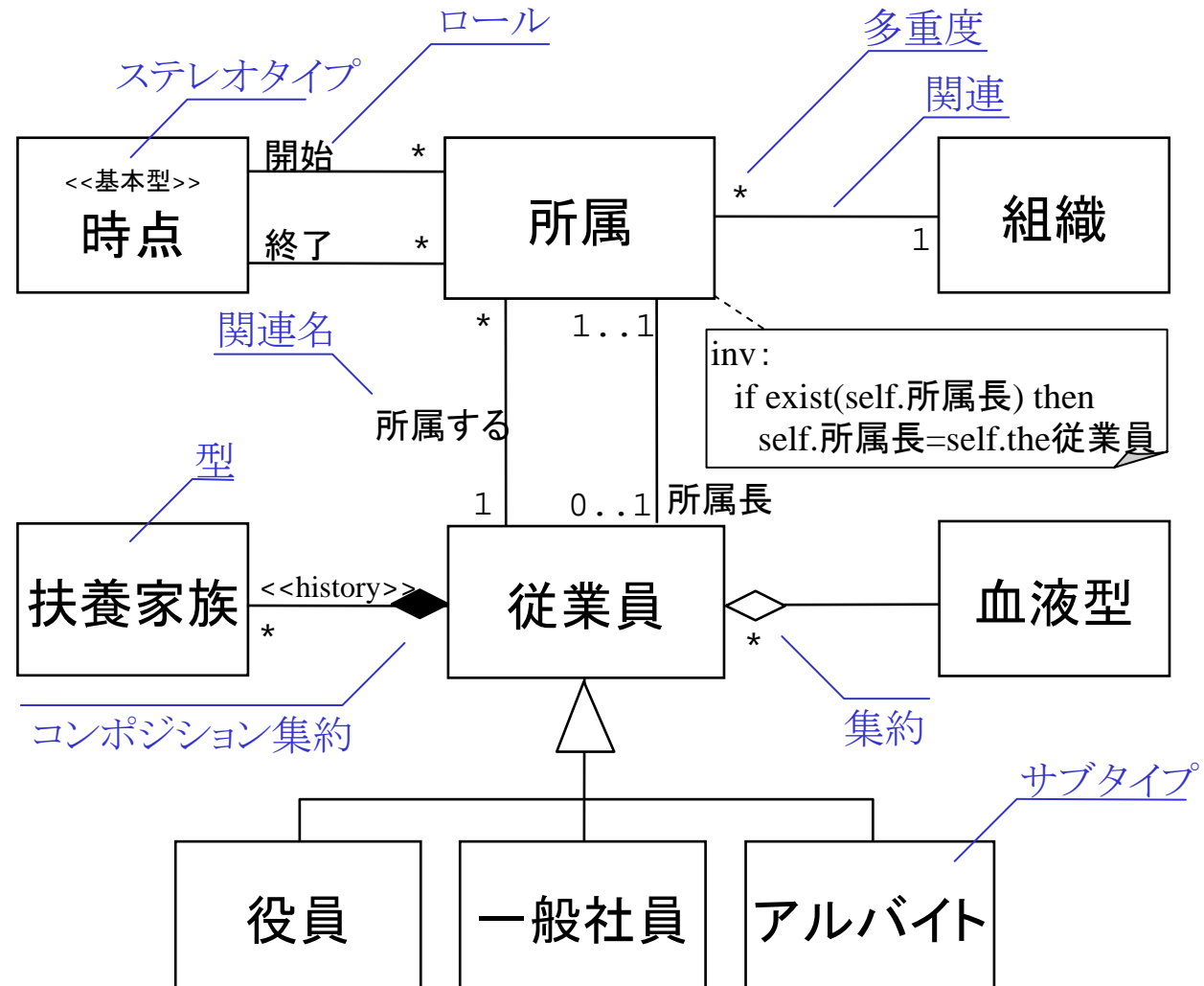
□ 導出



2.2 静的側面 (5)

□ 型図の例

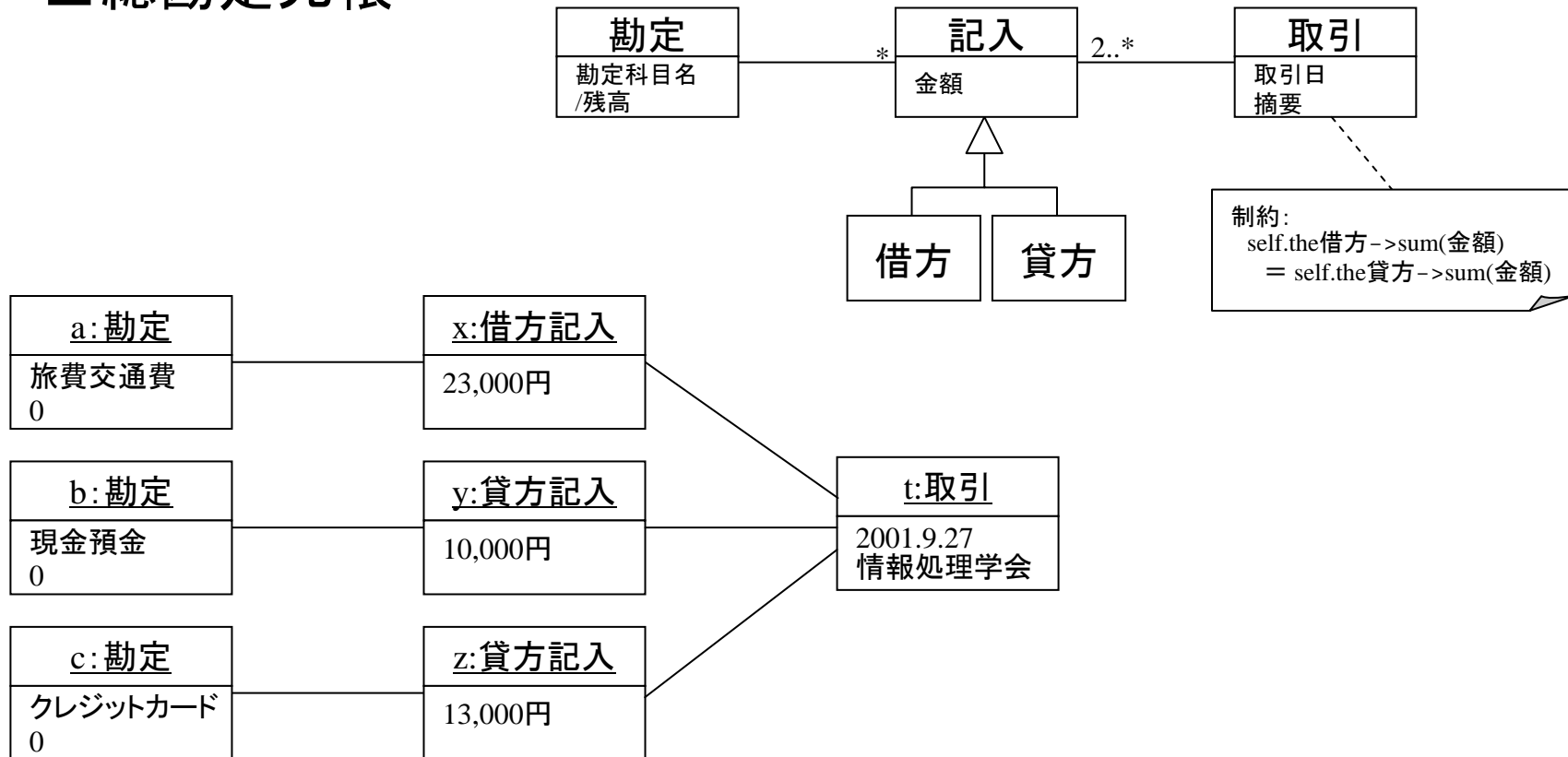
□ 従業員管理



2.2 静的側面 (6)

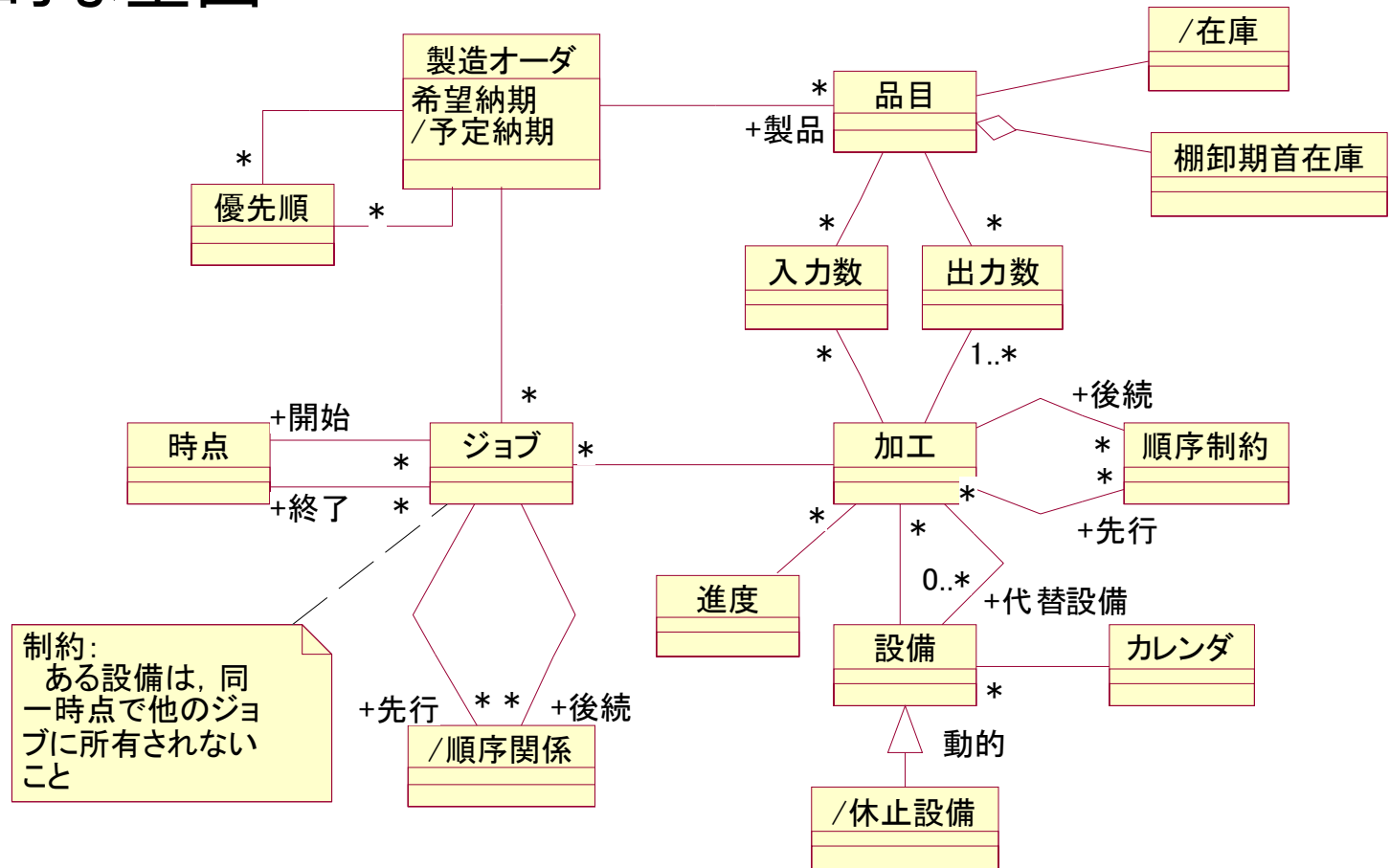
□ インスタンス図による理解

□ 総勘定元帳



2.2 静的側面 (7)

□ 典型的な型図



2.3 動的側面(1)

□ビジネスプロセス

- ビジネス目的を達成するための手順

- 一連のプロセスステップ

 - ビジネスコミュニケーション

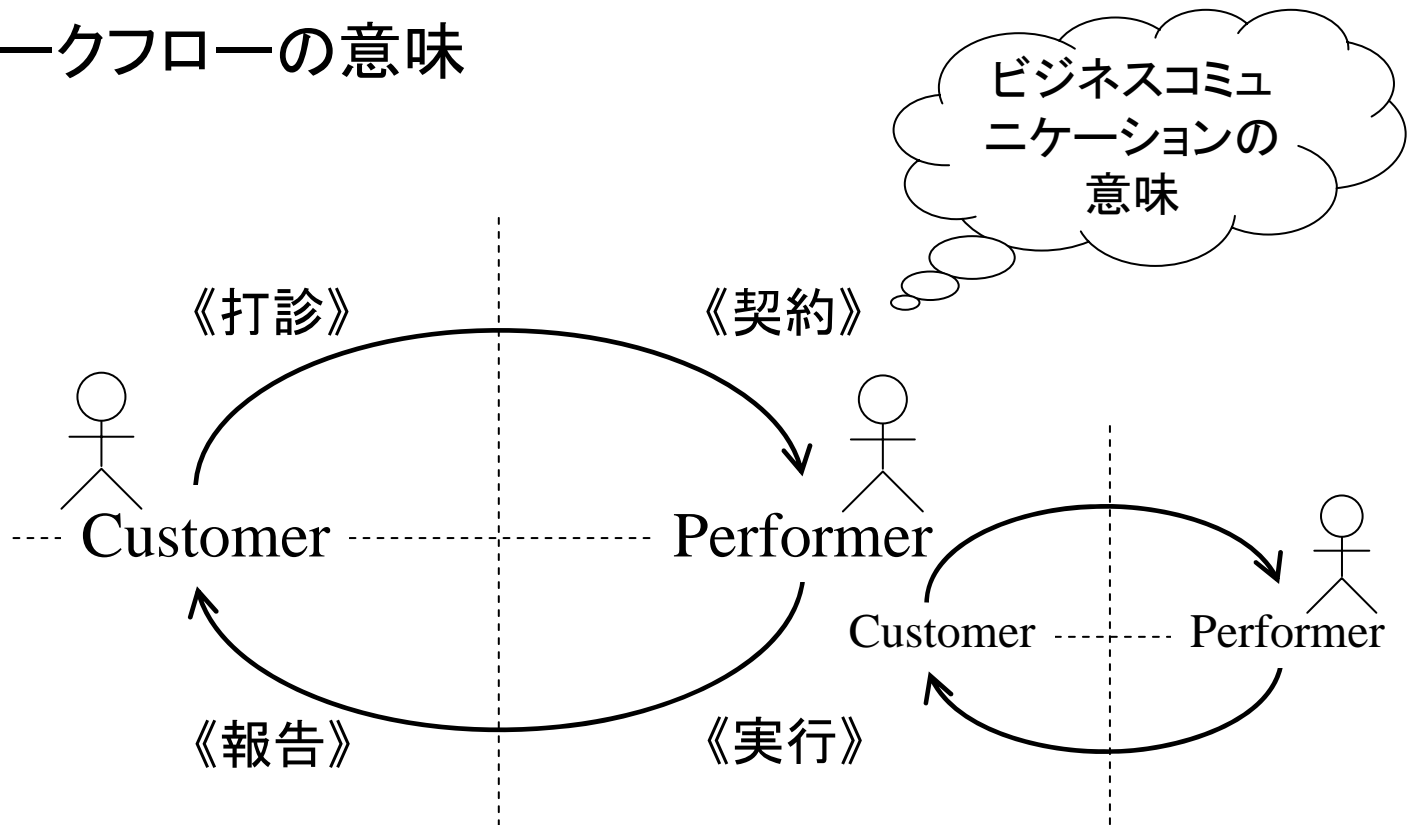
- プロセスごとのコスト

2.3 動的側面 (2)

□ Customer-Performer

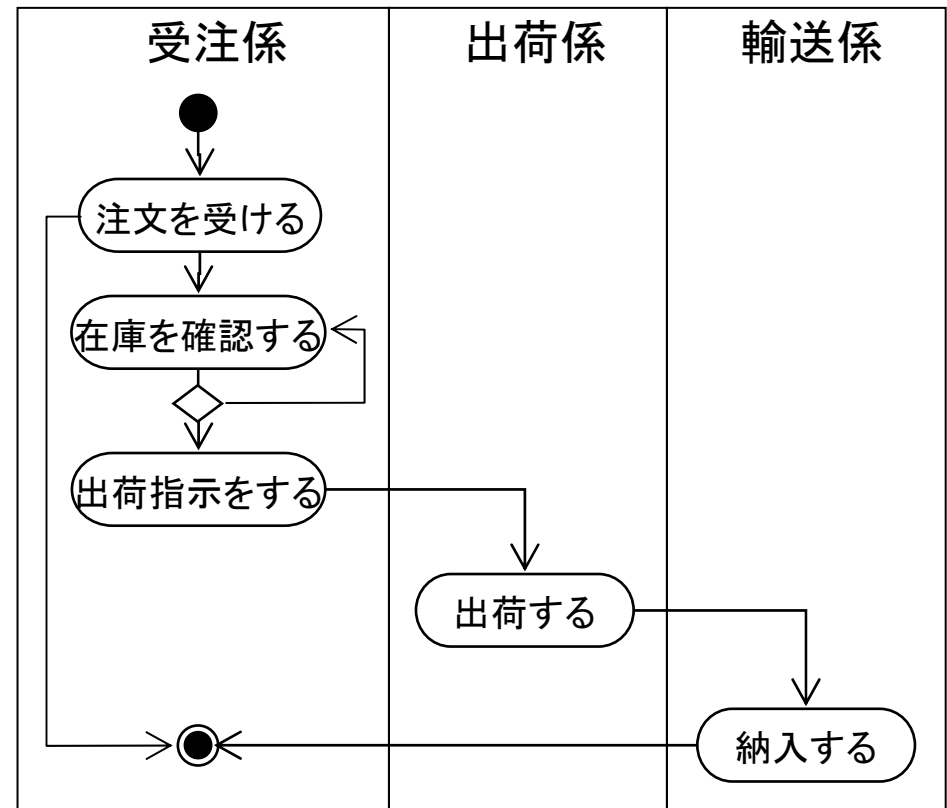
□ ビジネスプロセスの最小単位

□ ワークフローの意味



2.3 動的側面 (3)

- ワークフローの記述
 - アクティビティ図
 - ロールアクティビティ図



2.3 動的側面 (4)

□アクティビティ図で書かれないもの

□ソフトウェアに割当てられた／割当てられない要求

□アクタの活動

□情報システムの支援あり

□情報システムの支援なし



ユースケース

□状態遷移するオブジェクト

□システム外部でのコミュニケーション

□Customer-Performer

□コミュニケーションの媒体

2.3 動的側面(5)

□ユースケース記述の例

ユースケース名：顧客オーダーをチャージに詰め込む

目的：できるだけチャージにロス(製品在庫)が発生しないように、製品仕様の類似性の高い顧客オーダーを組み合わせて、チャージ編成計画を上流工程に要求する。

アクタ：厚板製造計画担当者

事前条件：詰め込む先のチャージ計画が存在する。詰め込むべき顧客オーダーが0件以上存在する。

手順：

1. アクタは、システムに顧客オーダーの詰め込み指示を行う旨を伝える。
2. システムは、詰め込む先のチャージ計画の選択を求める。
3. アクタは詰め込む対象のチャージ計画を指示する。
4. システムは、その計画に詰め込む顧客オーダーを求める。
5. アクタは、詰め込む顧客オーダーを指示する。手順4, 5を繰り返す。
6. システムは、アクタに詰め込み指示を記録したことを伝える。
7. アクタは、必要に応じて手順1から6までを繰り返す。

事後条件：詰め込み指示が記録されている。

代替系列：なし。

備考：

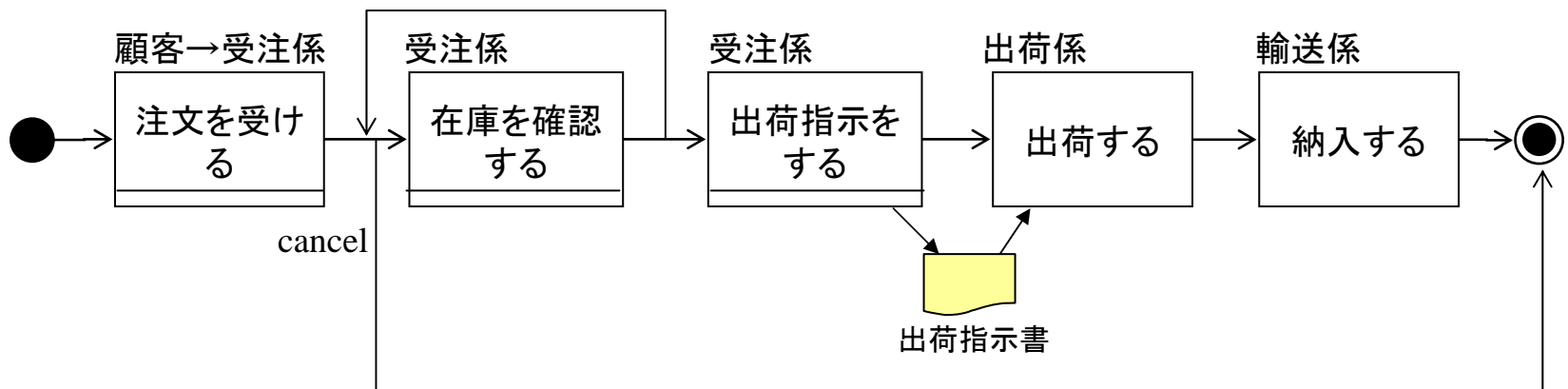
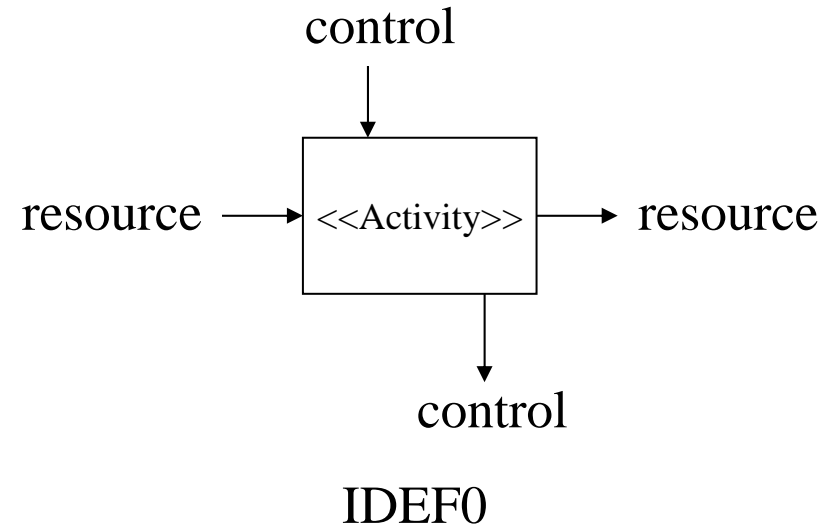
- (1) このユースケースによって製造作業の平準化ができるだけ実現されることが期待されるが、必ずしもそうなるとは限らない。
- (2) チャージ計画の内容は、{製造番号, 品目(諸元がすべて明確になっていなくてもよい), 数量, 完成予定日}とする。

2.3 動的側面 (6)

□ワークフロー図

□WfMC + Kodama

- アクティビティ(アクタの活動)
- システム外のアクタとの通信
- 実行コスト(所要時間, 必要スキル)

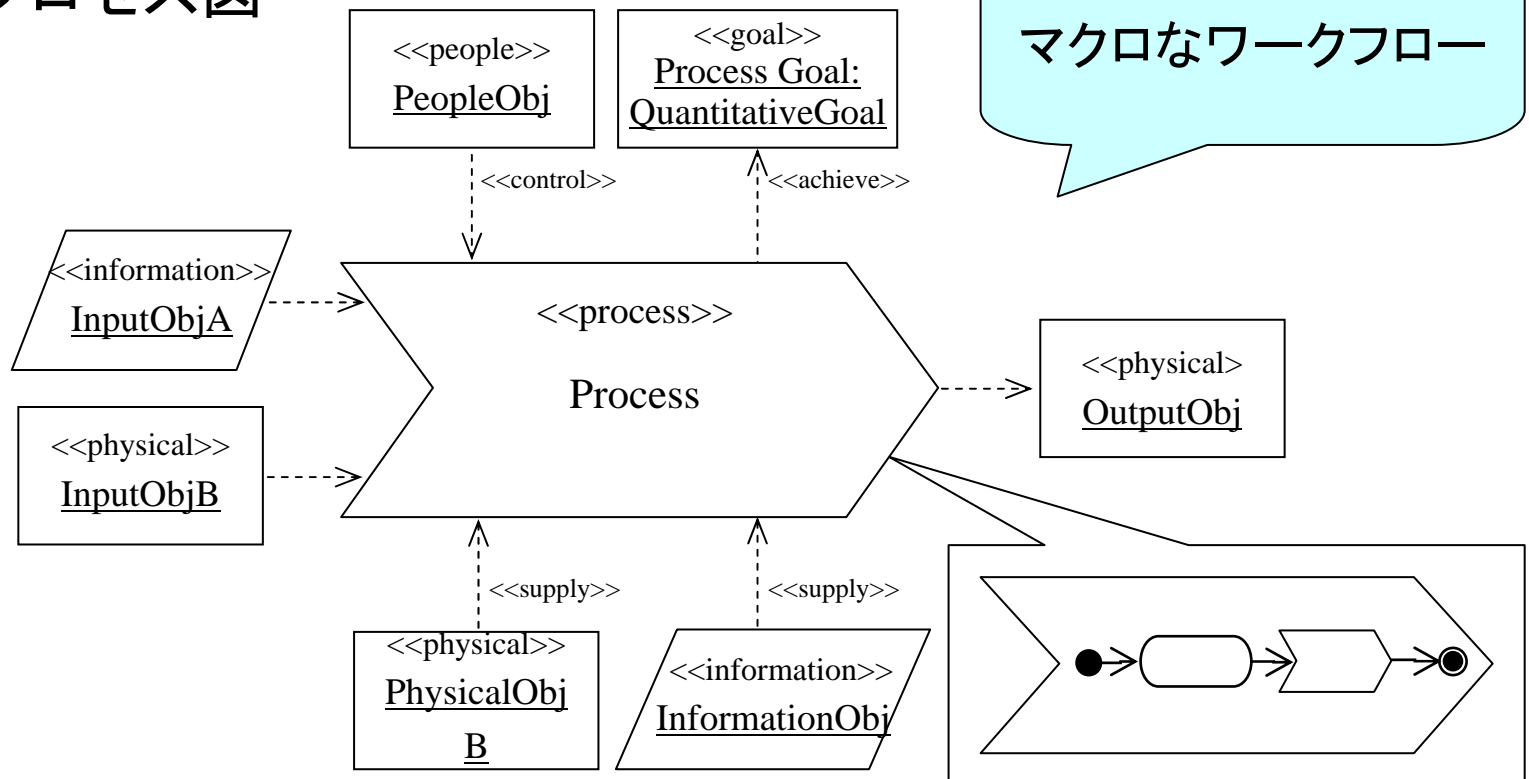


2.3 動的側面 (7)

□ Eriksson & Penkerの拡張

□ アクティビティ図の拡張

□ プロセス図



2.3 動的側面 (8)

□ ATO (Assemble to Order)

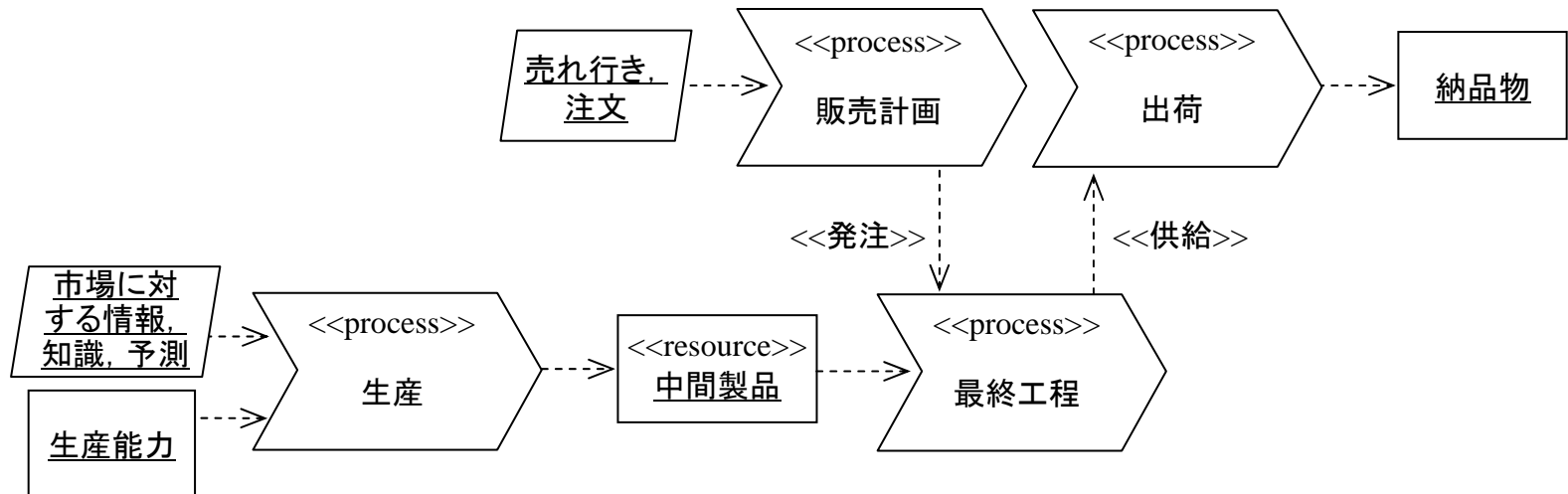
□ B社のビジネスモデル

□ 製品在庫を最小化

□ 中間製品での在庫

□ 染色を最終工程に

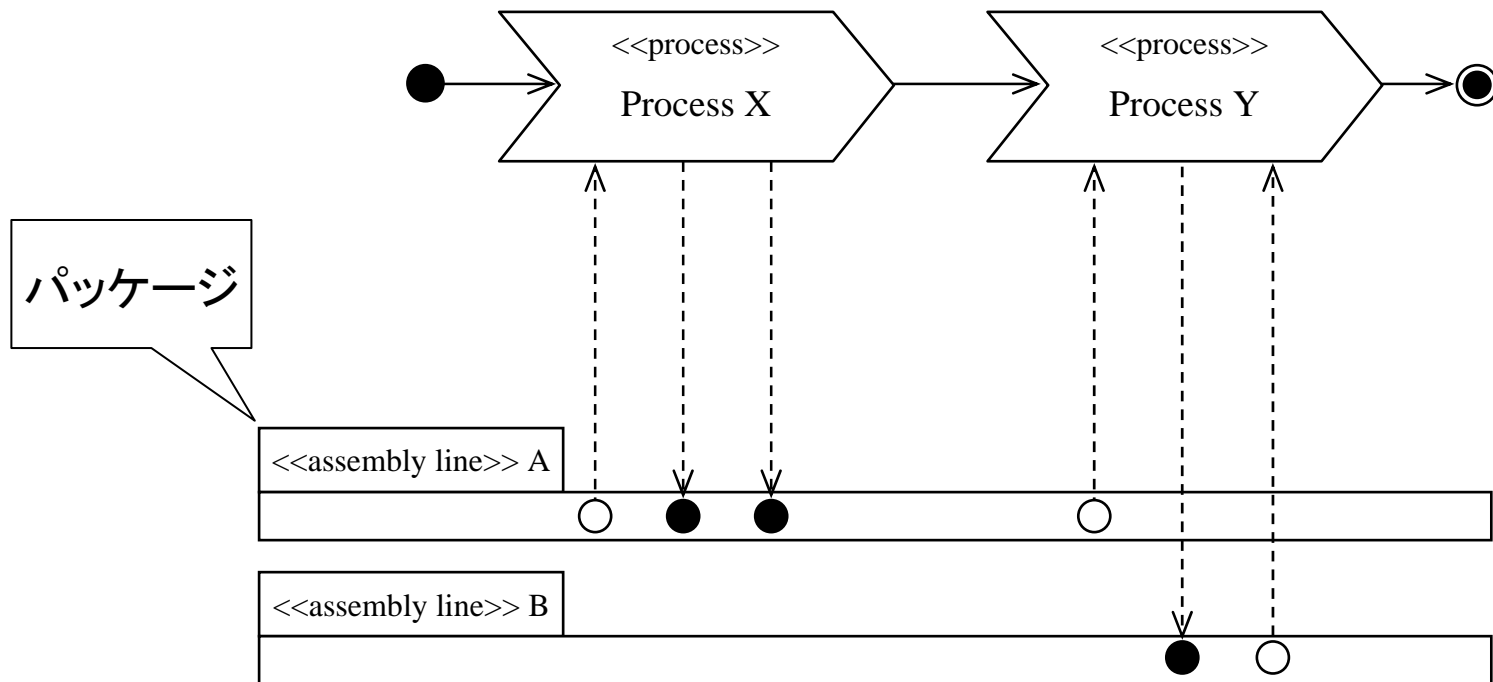
□ (オプションを最終工程で実装する)



2.3 動的側面 (9)

□ Eriksson & Penkerの拡張

□ 組立ライン図

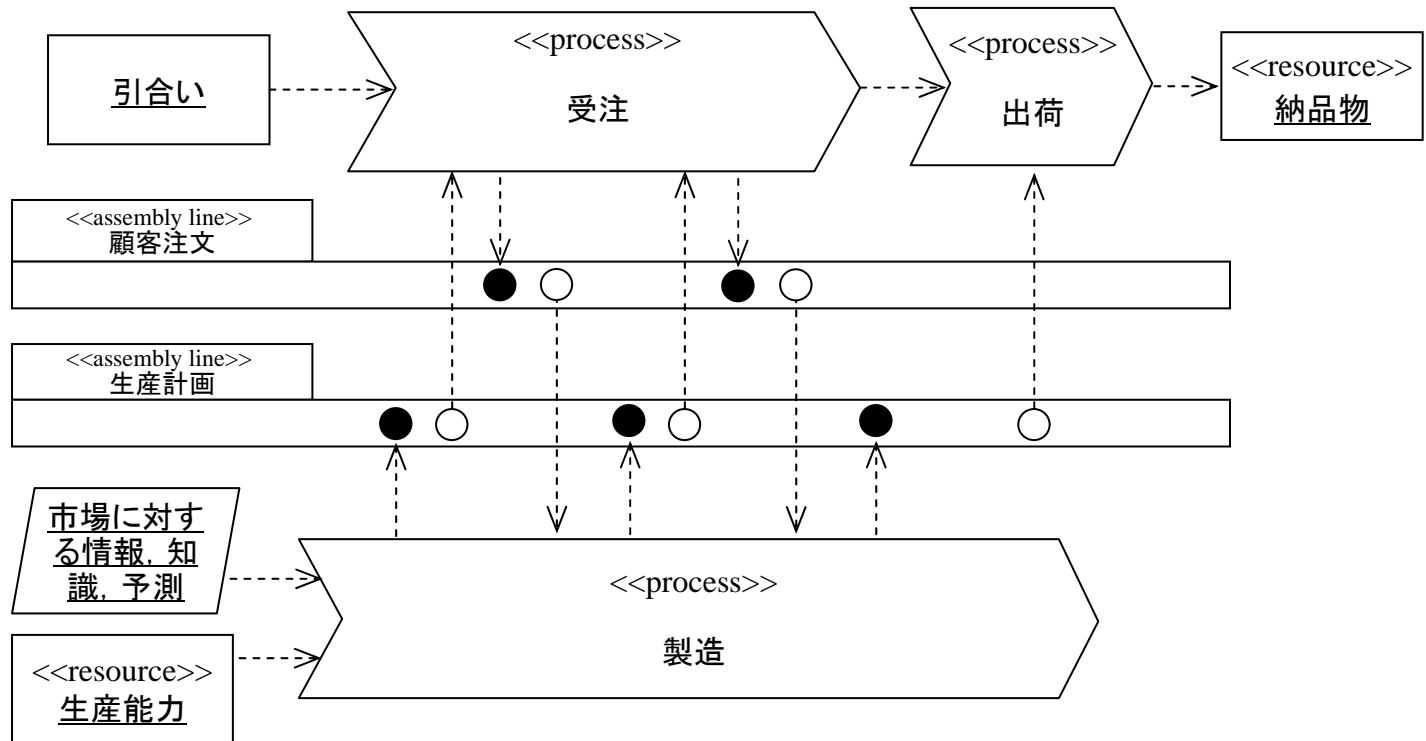


2.3 動的側面 (10)

□ 生産座席予約

□ 製造リードタイムの短縮と設備使用の活用

□ 先行的に製造(座席)しながら, 顧客注文(乗客)を引き当てる



2.4 静的側面と動的側面の整合

- 静的側面(ドメイン)は広めに書く
- 動的側面は狭く
 - 書かれたユースケースが, そのドメインで動くように補正していく
 - ドメイン→ユースケース→ドメイン
 - 型とそのインタフェースの検討
 - 主要なメソッドの検討
 - 主要な属性の検討

ユースケースがないと検証できないが, ユースケースだけあっても何もできない

3. 何をモデル化するか

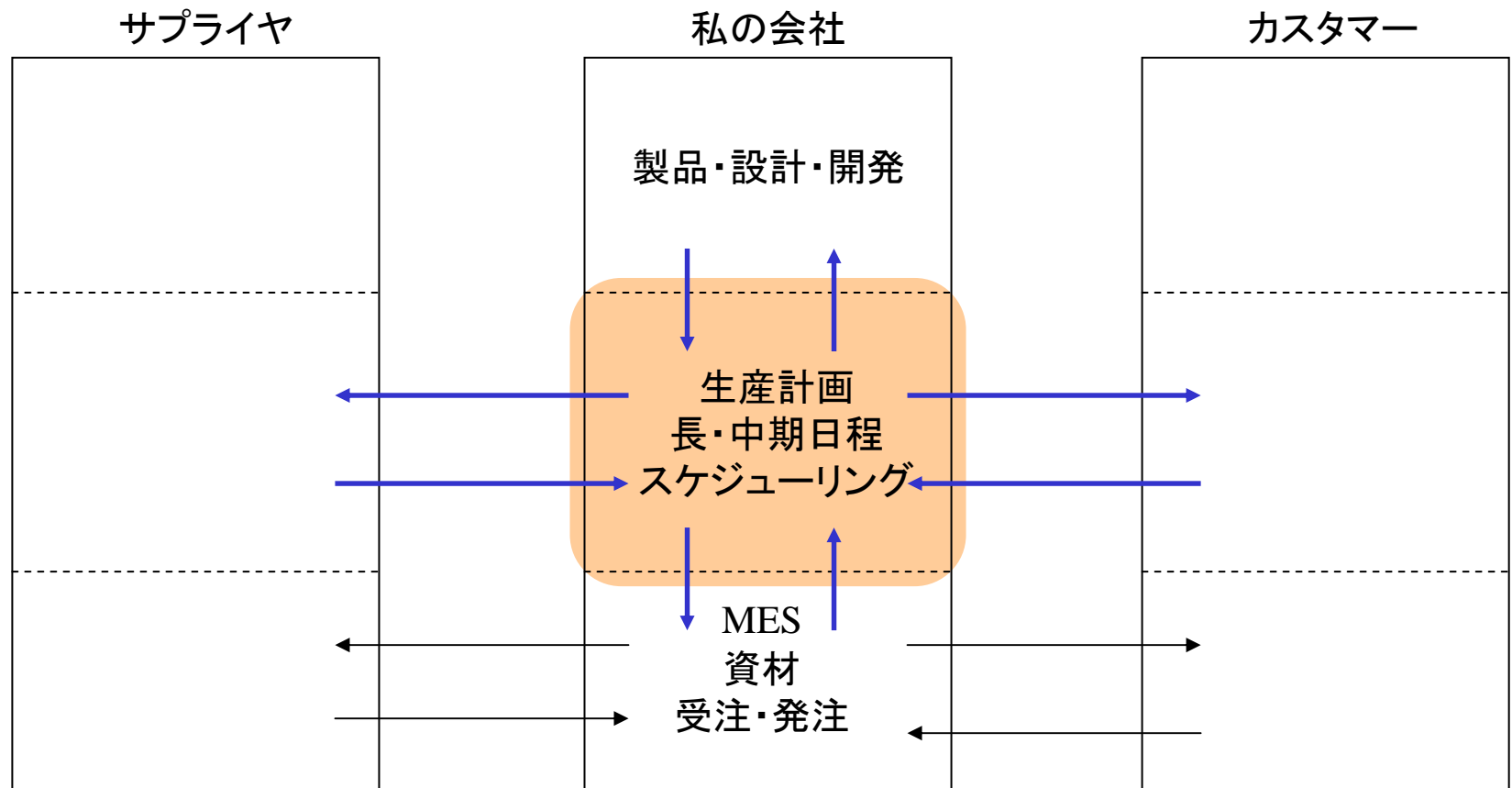
3.1 目標

3.2 Embrace Change

3.3 改革を阻むもの

3.1 目標(1)

□モデルの範囲



3.1 目標 (2)

□メンバが同意可能なオブジェクトモデルを書く

□モデルの内容

□現状のスケジューラの機能を網羅？

□あるべきスケジューラの機能を取り込む？

□シナリオを書いてユースケースを抽出

□典型的なドメインモデルと突き合わせて改訂

□モデルの形式

□XMLに変換可能であること

□実装まではコミットしない

□アルゴリズムには言及しない

日本発信の新しい生産管理システムを構築するんじゃないか？

3.2 Embrace Change

□ 変更を抱擁せよ

- 注文仕様の変更

- 生産計画の変更

- 頻繁な設計変更

□ 日本型マス・カスタマイゼーション実現のために

- 見込みで製造を始め、途中の変更を受け入れて、顧客オプションを実現していく

 - 製造リードタイムの短縮

 - タイムリーな市場投入

3.3 改革を阻むもの

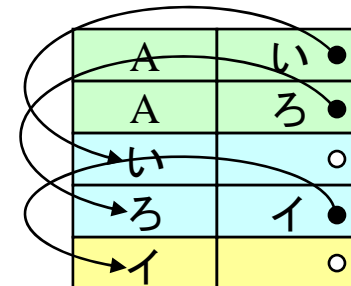
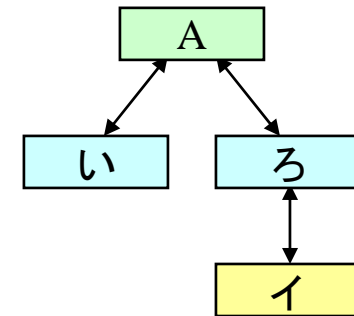
□ 品目コード

- 変更を許容しにくい原因
- コードだけでは類似性を扱えない
- ファントム部品を強制

□ RDB

□ BOM部分

- 品目コードを強制
- 値によるナビゲーション



4. ロードマップ

4.1 ロードマップと進捗

4.2 他分科会との調整

4.1 ロードマップと進捗

9/19 スケジューラ利用のシナリオ検討, ユースケースの抽出

10/25 ユースケース記述の作成

グランドデザイン分科会との調整, XML分科会との調整

11/30 オブジェクトモデルの作成, ユースケース記述(残)の作成, クラス図の作成, アクティビティ図の作成

12/20 オブジェクトモデルの作成, <<interface>>の明示

XML分科会との調整

1/24 まとめ(案), 文章の作成

3/x ワーキングドラフト一般公開

4.2 他分科会との調整

□ グランドデザイン分科会

- 日本発信の新しい生産管理システム中での位置づけ
- スケジューラに必要な機能

□ 統一用語辞書分科会

- 概念の整理

□ XMLとインターネットコア技術分科会

- オブジェクトモデルのXML化可能性

まとめ

なぜオブジェクトモデルなのか

- インタフェースそのものの定義

- 目に見える形で

オブジェクトモデリングの方法

- ドメイン→ユースケース→ドメイン

何をモデル化するか

- 新しい生産管理システム？

ロードマップ