

ビジネスからSOAへ

細川 努

SOAとは

SOAについての技術動向

- SOAによって、今、何ができるのか

要求開発と**SOA**

SOAとは

セミナーとかに行ってもいろいろ話が出てきますが...

▪

BPMN

WSDL

Webサービス

ワークフロー

**ソフトウェア
コンポーネント**

いずれも**SOA**を実現する要素

SOAって結局なんでしょう？

SOA (Service Oriented Architecture)

- ・ソフトウェア機能をサービスとして利用できるようにする
- ・サービスを標準的なインタフェースから利用できる

サービス同士の連携として機能できるシステムアーキテクチャ

SOAでないシステム

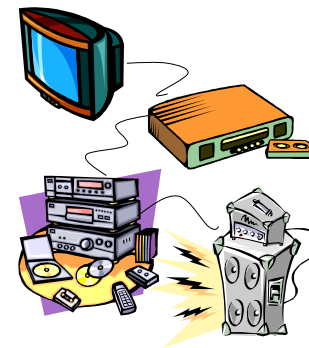
AV機器
に例える
なら...



他のサービス（機器）と
簡単に接続できない

壊れたら買い替え

SOAなシステム

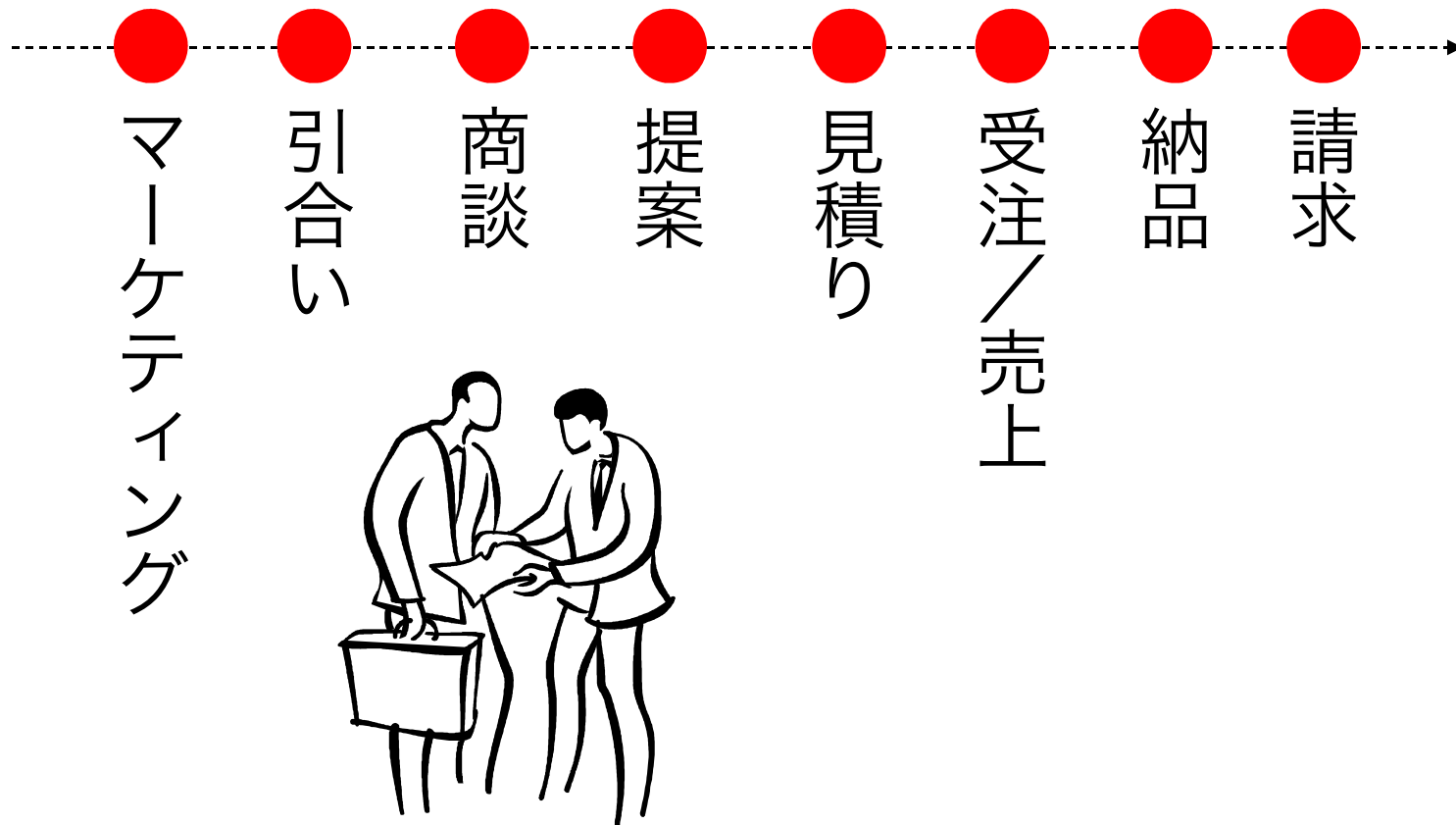


サービス同士の**インターフェース**を
接続すると連携して使える

必要に応じて組み替え

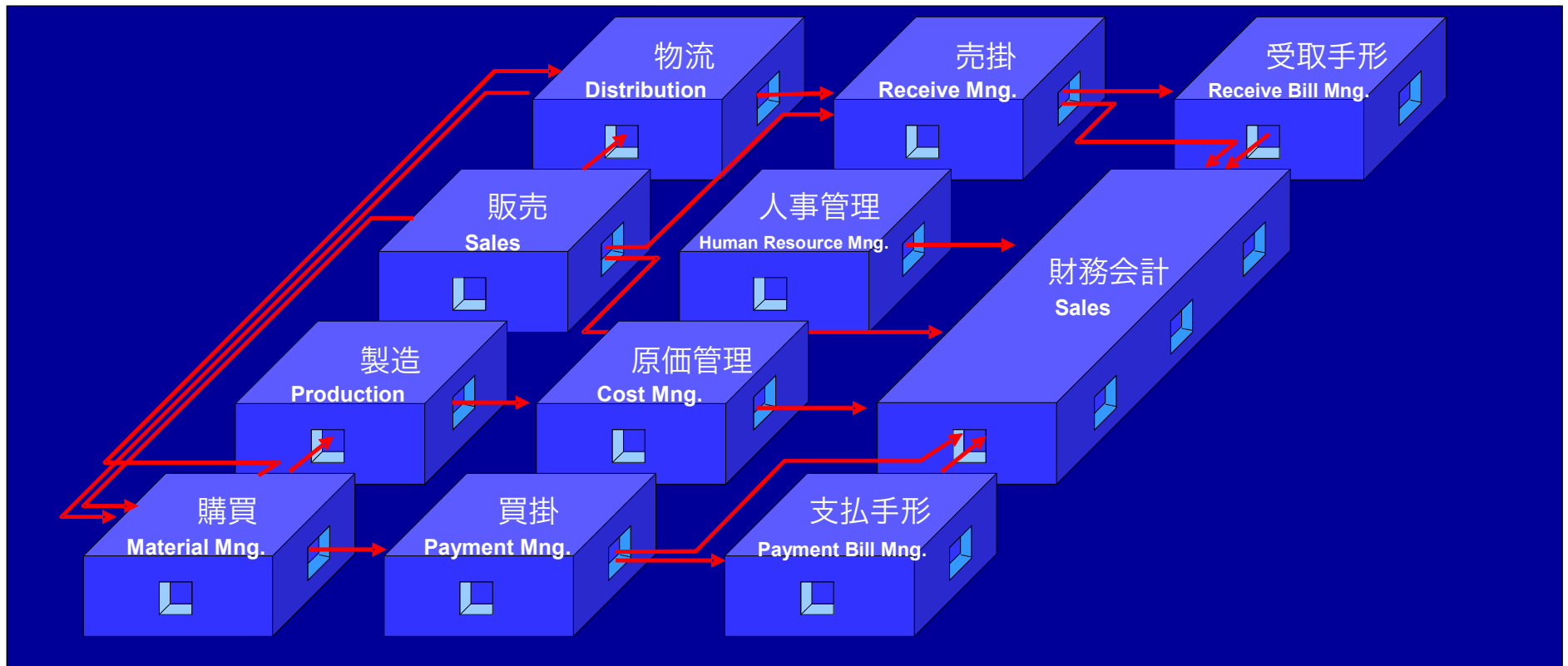
ビジネスプロセス

例えば商品を販売する時には、一連のプロセスがある。



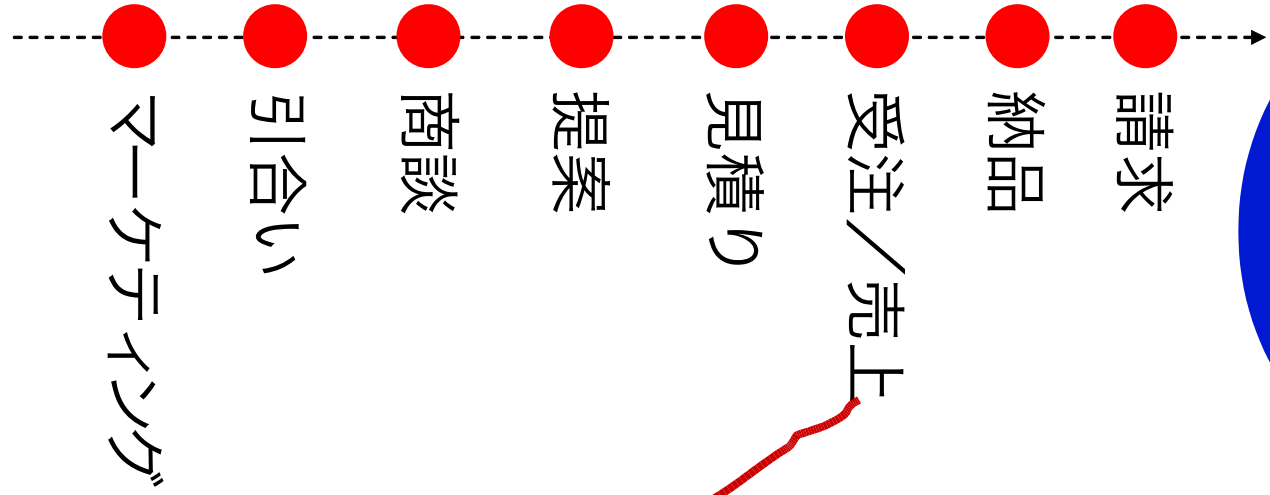
システムのビジネス機能

システムのビジネス機能同士も複雑に絡み合っている



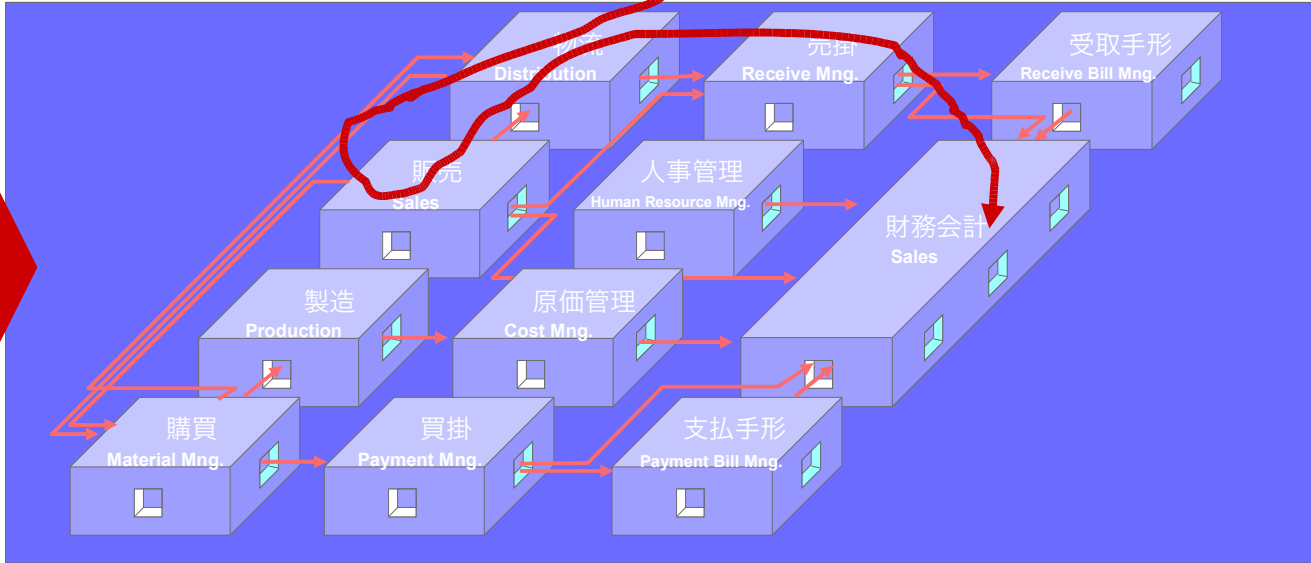
ビジネスプロセスとシステムサービス

ビジネス
プロセス



ワーク
フロー

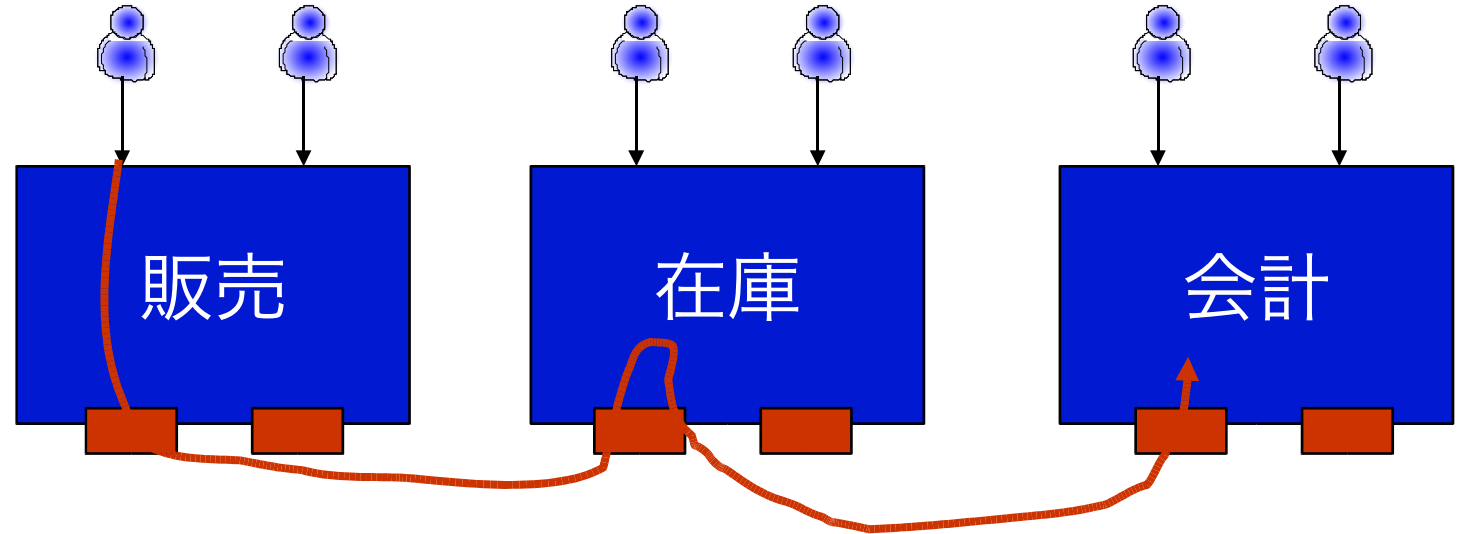
システム
サービス



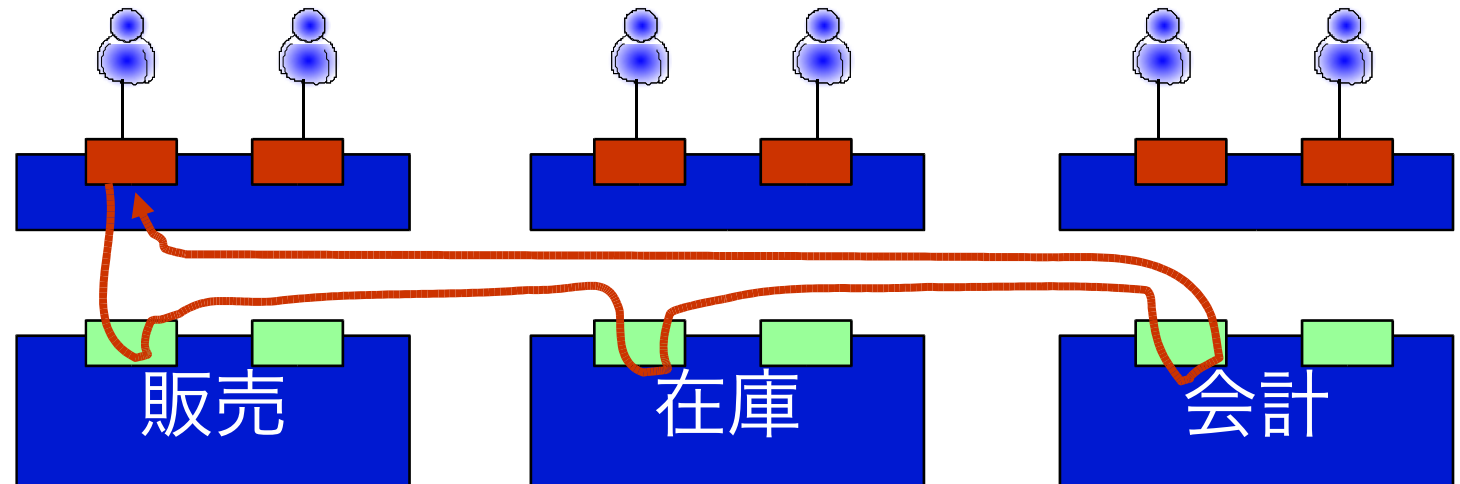
サービス
アーキテクチャ

SOAシステムアーキテクチャ

① バックヤードアーキテクチャ

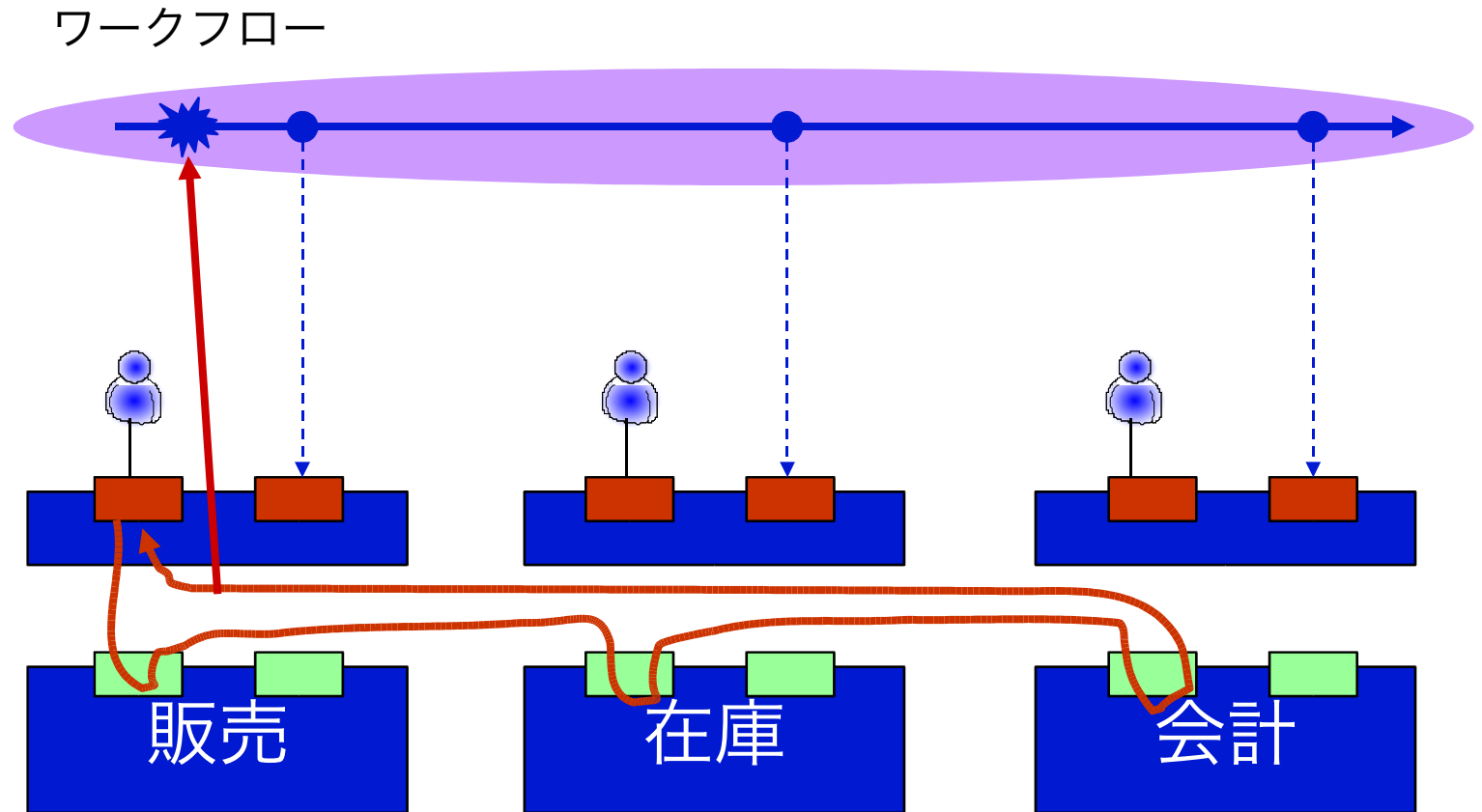


② コンポジットサービスアーキテクチャ

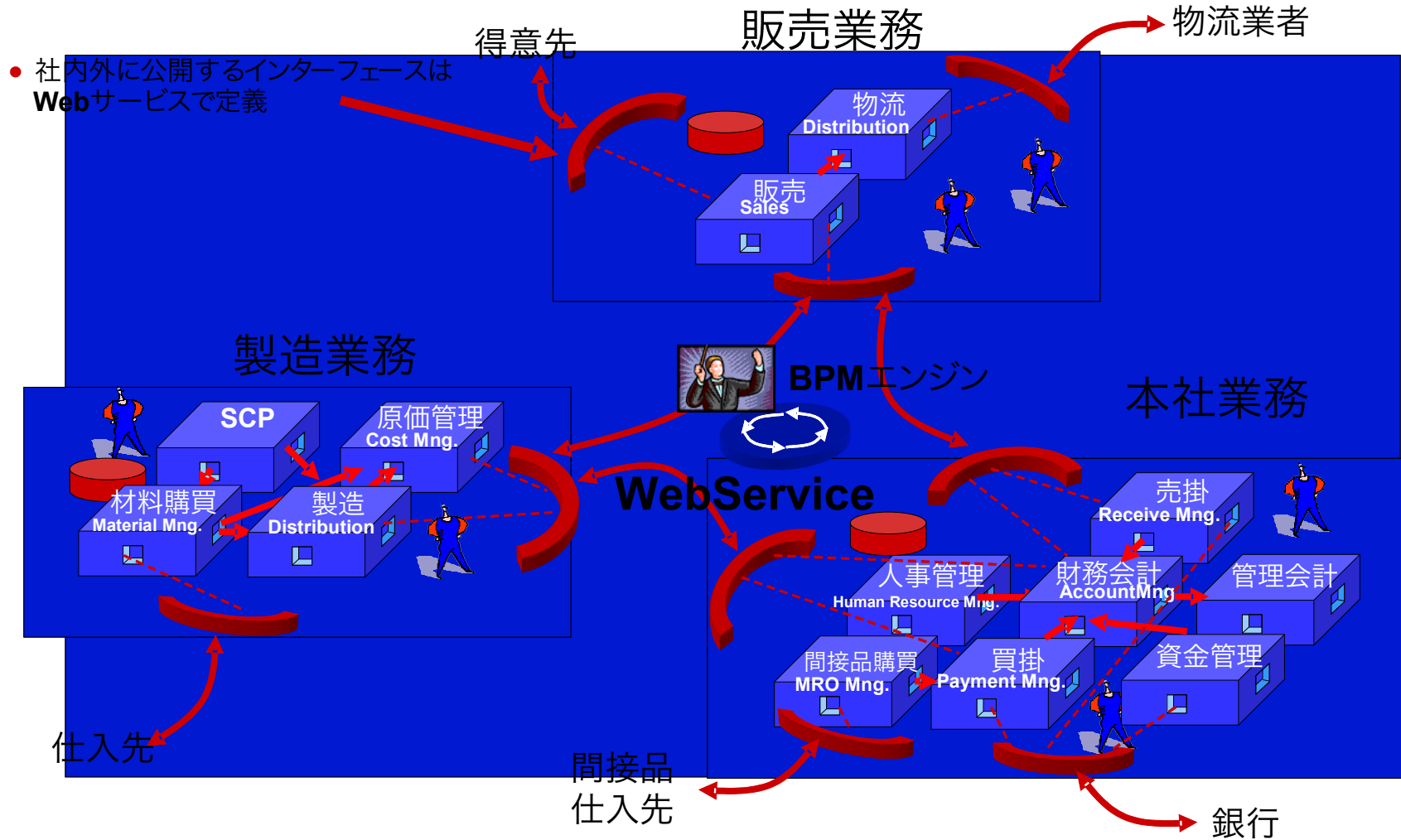


SOAシステムアーキテクチャ

③ サービスワークフローアーキテクチャ



SOAアーキテクチャの例



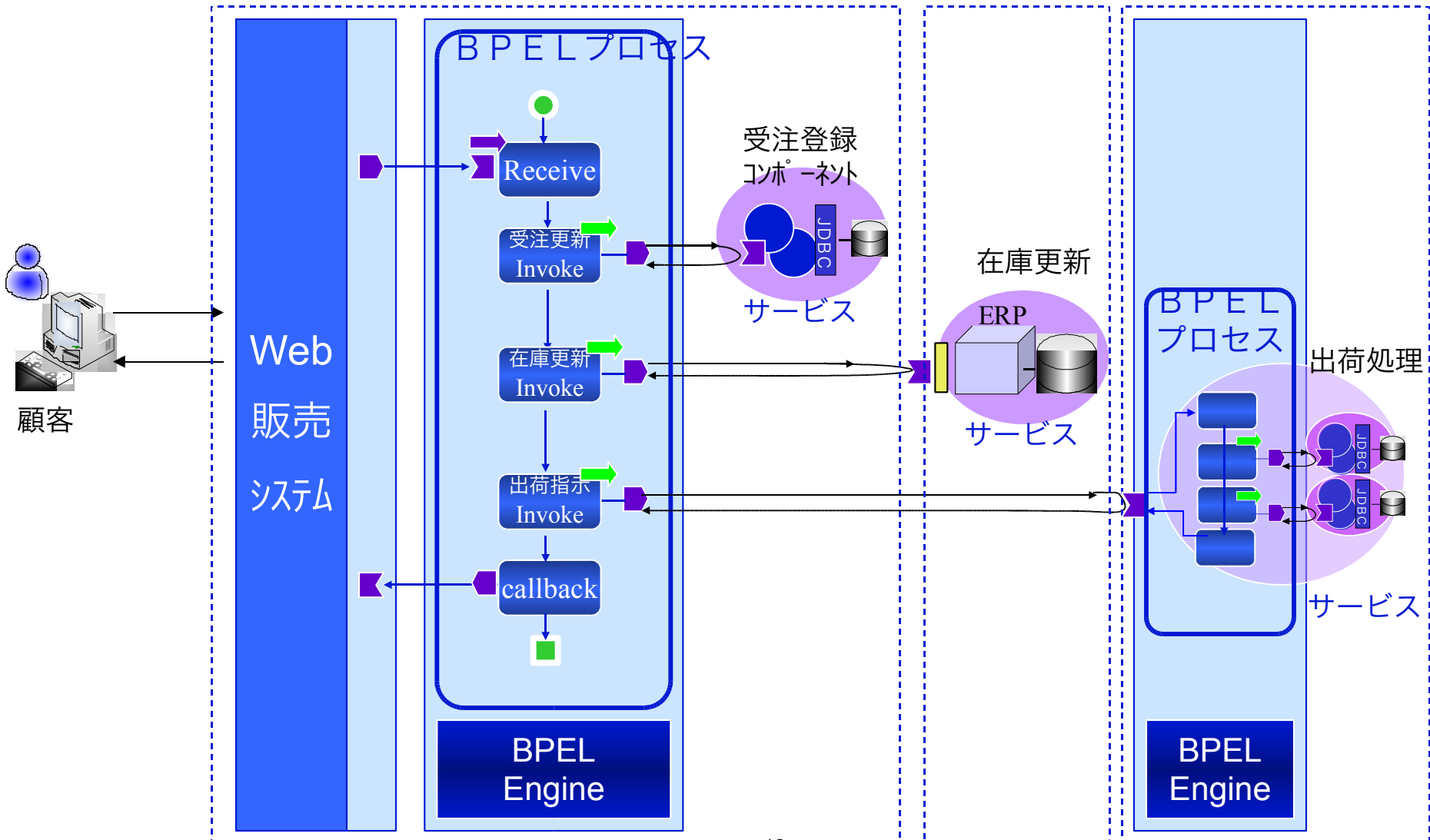
BPEL

The screenshot displays the Oracle JDeveloper 10g IDE with a BPEL process diagram open. The main workspace shows a flow diagram for `myCreditFlow.bpel`. The process starts with an `assignSSN` activity, followed by an `invokeCRS` activity that calls the `creditRatingService`. This is followed by an `assignCreditRating` activity. The diagram is contained within a `Sequence_1` container. On the right, the `Component Palette` shows process activities like `Assign`, `Compensate`, `Empty`, `Flow`, and `FlowN`. Below it, the `Assign_1 - プロパティ` window shows the `joinCondition` property set to `Assign_1`. The `Project Navigator` on the left shows the project structure, including `myCreditFlow.bpel`. The `myCreditFlow.bpel - 構造` window at the bottom left shows the activity structure, with `アクティビティ構造` selected. The status bar at the bottom indicates the file path and `Map file editing`.

BPEL

A社（販売管理）

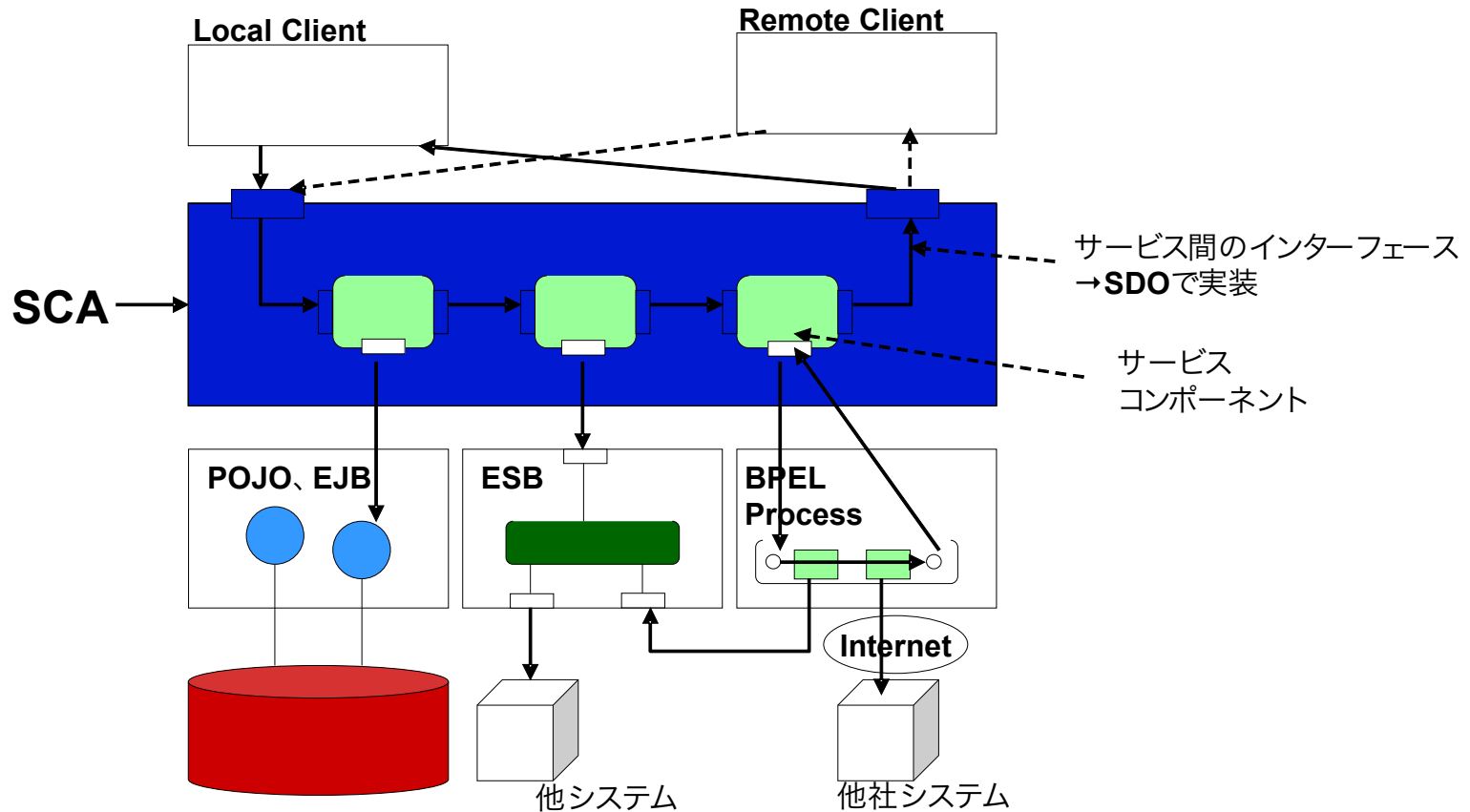
アウトソース先
B社（在庫管理）C社（物流管理）



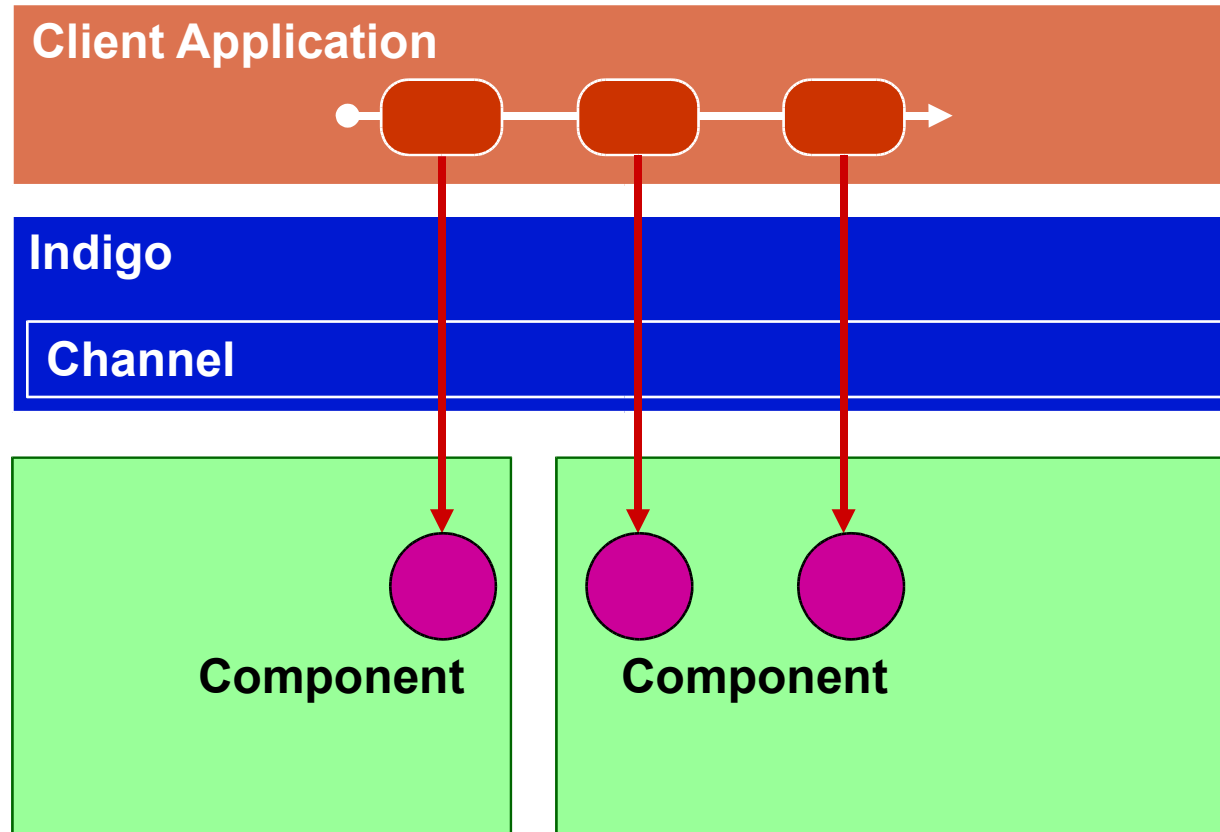
SCA (Service Component Architecture)

SCA

SCA :Service Component Architecture
SDO :Service Data Objects



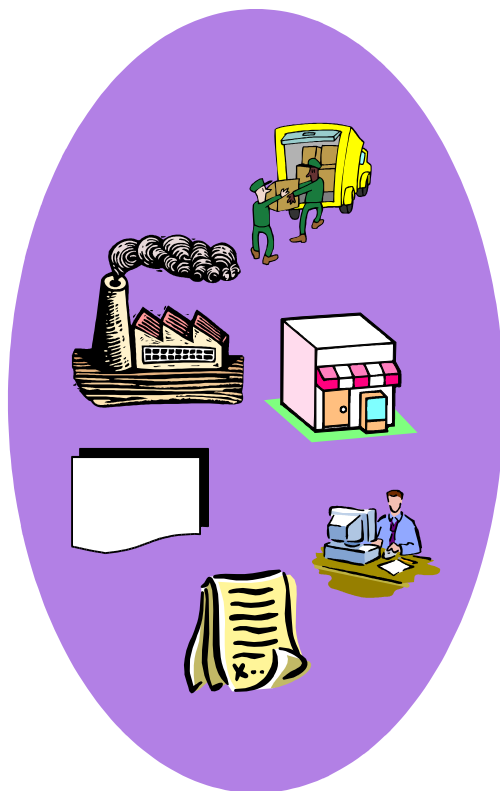
Microsoft Indigo



でも、どうしたらビジネス要求をSOAにつなげられるの？

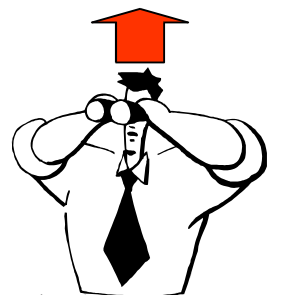
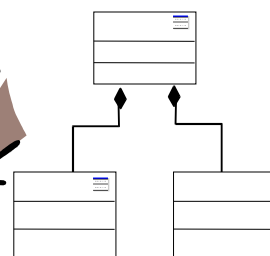
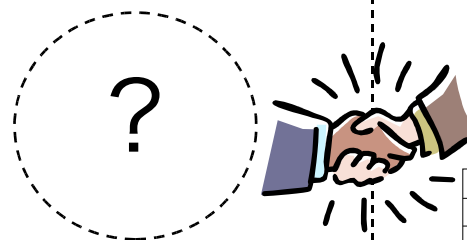
ビジネス

システム

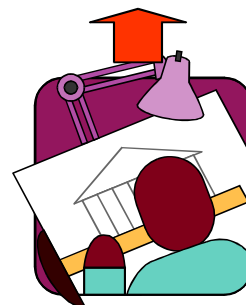


ビジネス設計

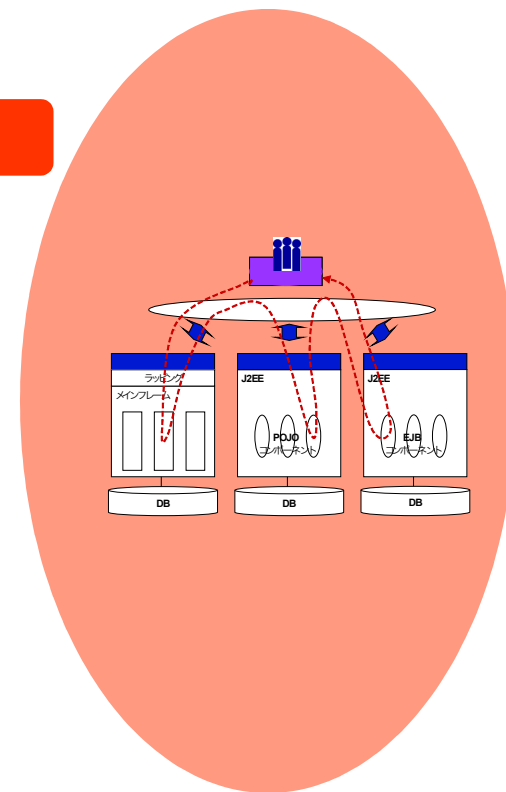
システム設計



ビジネスで扱う
情報の鳥瞰図



システムとしての
設計図



そこで要求開発方法論の出番です！

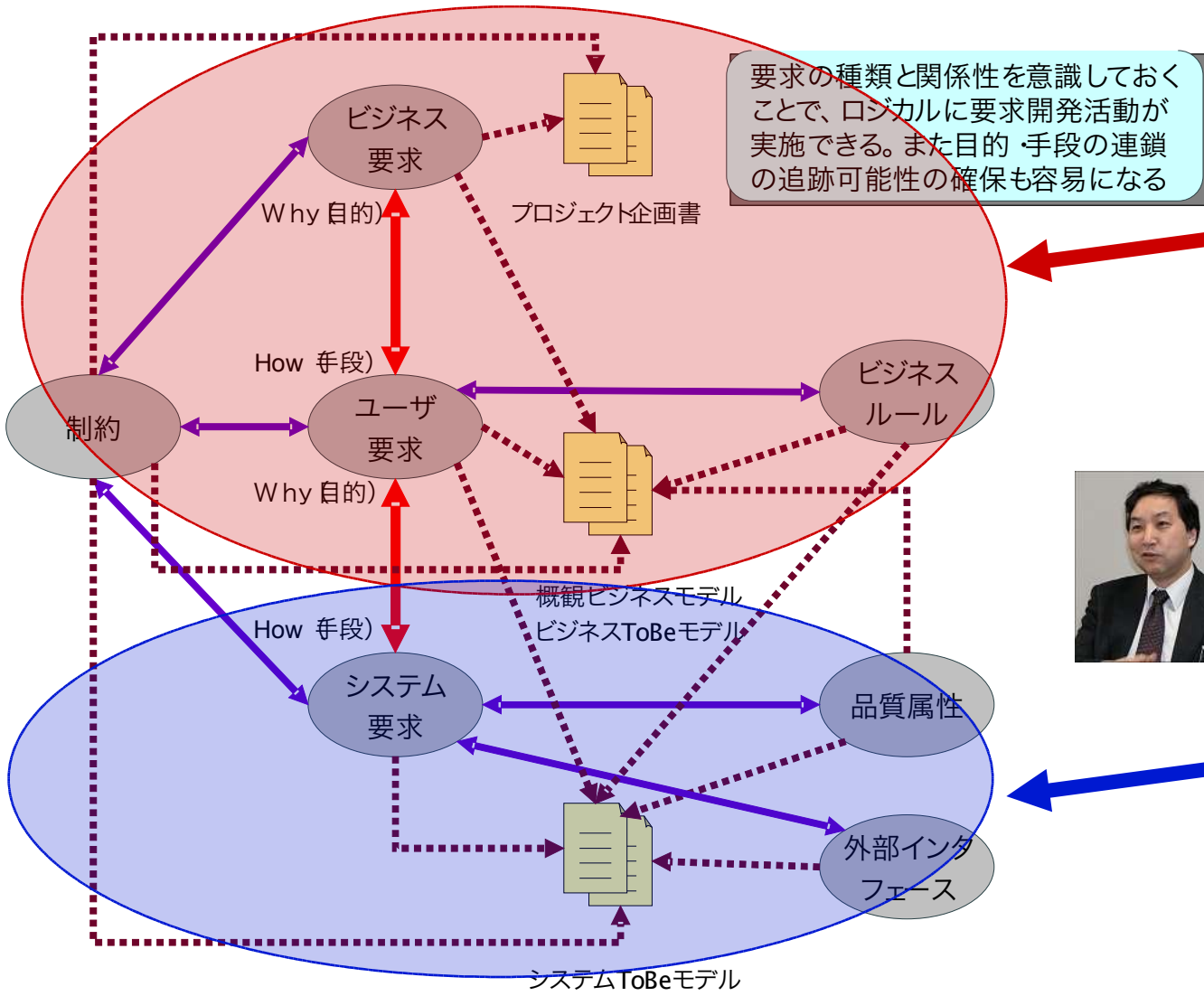


日経BP社
ISBN: 4822282686 (2006/03/02)



翔泳社
ISBN: 4798110485 (2006/03/07)

要求開発とは



要求の種類と関係性を意識しておくことで、ロジカルに要求開発活動が実施できる。また目的・手段の連鎖の追跡可能性の確保も容易になる

ビジネス要求
を「開発」する

要求は「在る」ものではなく
「作る」ものでおじゃる



要求開発アライアンス

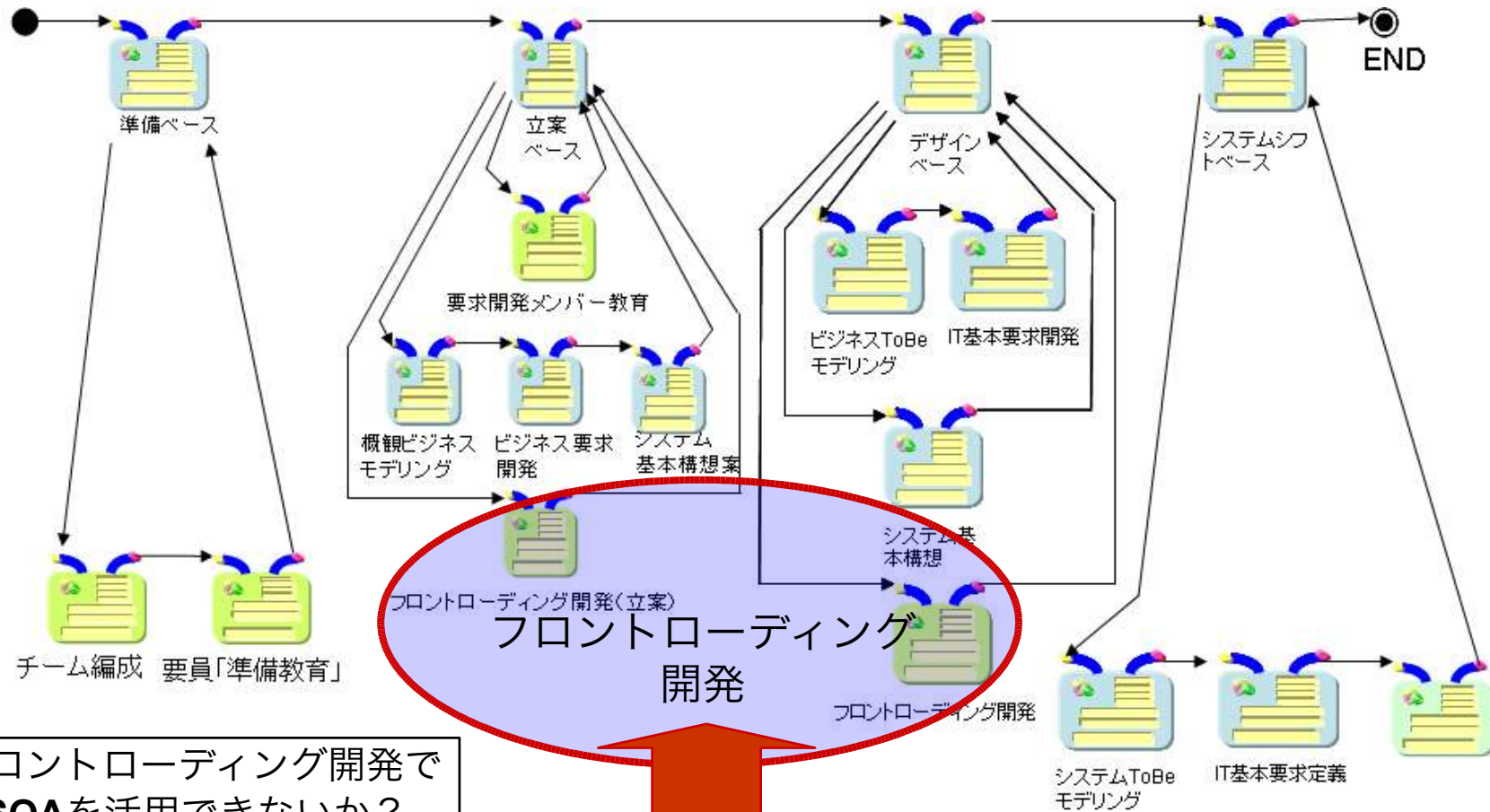
山岸理事長

システム要求

要求開発のプロセス

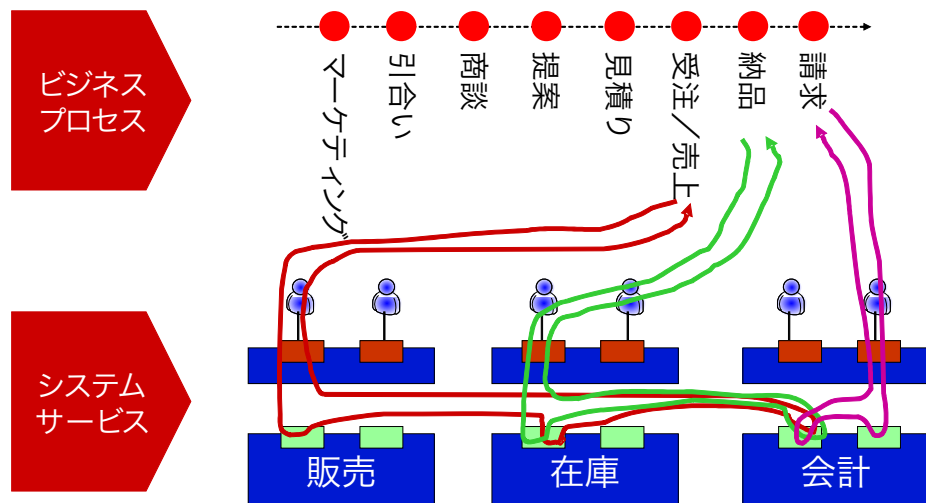
ビジネスの要求の開発

+ 要求を具体化するための試作システム開発（フロントローディング）



フロントローディング開発で
SOAを活用できないか？

ビジネスプロセスとSOAとモデリング



ビジネスとシステムを
またがった
プロセスを
どう設計するか

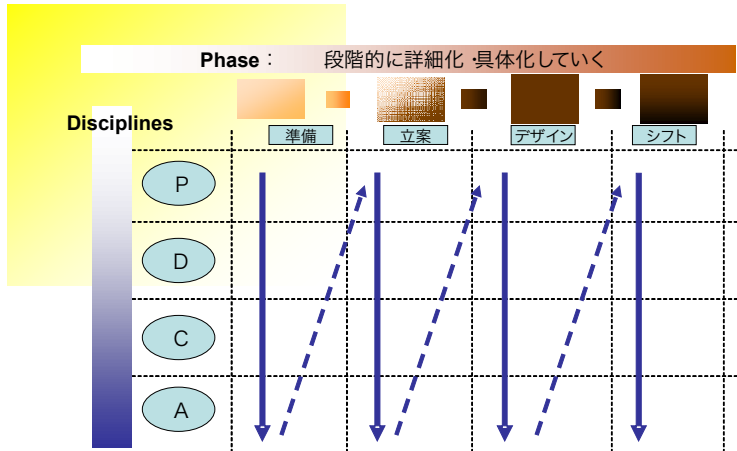
?

ここが今日の重要ポイントですね

Openthologyの概要

プロセス編 (作業の進め方)

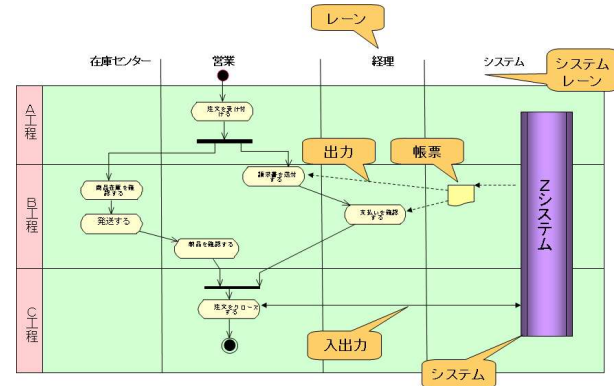
- プロセス (反復型PDCAサイクル)



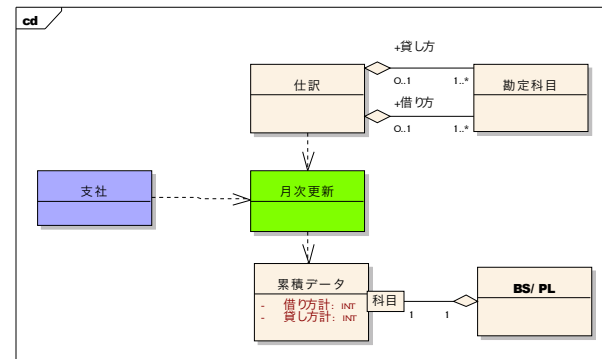
| Phase | 準備 arrangements | 立案 draft | デザイン design | システム開発への移行 |
|--------|--|---|--|---|
| Plan | フェーズ基本計画 事業計画プロジェクトのゴールの設定 参加開発チーム編成計画 研修教育計画作成 | フェーズ基本計画 モデルリング概観 ゼンレス要求評価方法の策定 教育計画の作成 ビジネス可視化プロトタイプ計画 システム基本構築策定計画 | フェーズ基本計画 ゼンレスモデルの評価方法の策定 基本要求の評価方法の策定 ビジネス可視化プロトタイプ計画の見直し システム基本構築策定計画 | フェーズ基本計画 ゼンレスモデルの評価方法の策定 基本要求の評価方法の策定 システム移行計画書作成/評価基準の明確化 |
| Do | 計画 ビジネスモデル 業務計画書作成 | 概観ビジネスモデリング | ゼンレスモデルモデリング | システム移行計画書作成 |
| | 要求モデル 教育 システム化 | ビジネス要求の調査 要求開発スパー教育 | ゼンレス要求の調査 基本要素の調査 | システム移行計画書作成 システム基本構築策定計画 |
| Check | 承認情報系による検証 メンバー意識・スキルの確認 教育実施評価 | 承認情報系による検証 プロジェクトレビューの評価 モデル概観評価 概観ビジネスモデリング評価 ゼンレス要求策定評価 教育実施評価 プロトタイプ評価 システム基本構築策定評価 | 承認情報系による検証 プロジェクトレビューの評価 モデル概観評価 プロトタイプ評価 システム基本構築策定評価 | 承認情報系による検証 システム移行計画書内訳の評価 システム移行計画書内訳の評価 |
| Action | 計画の見直し改善 メンバー意識の再確認 教育計画の再計画 必要であればビジネス課題の理解のやり | 計画の見直し改善 構想の見直し改善 不適合の修正と訂正改善 | 計画の見直し改善 構想の見直し改善 | 計画の見直し改善 |

モデル編

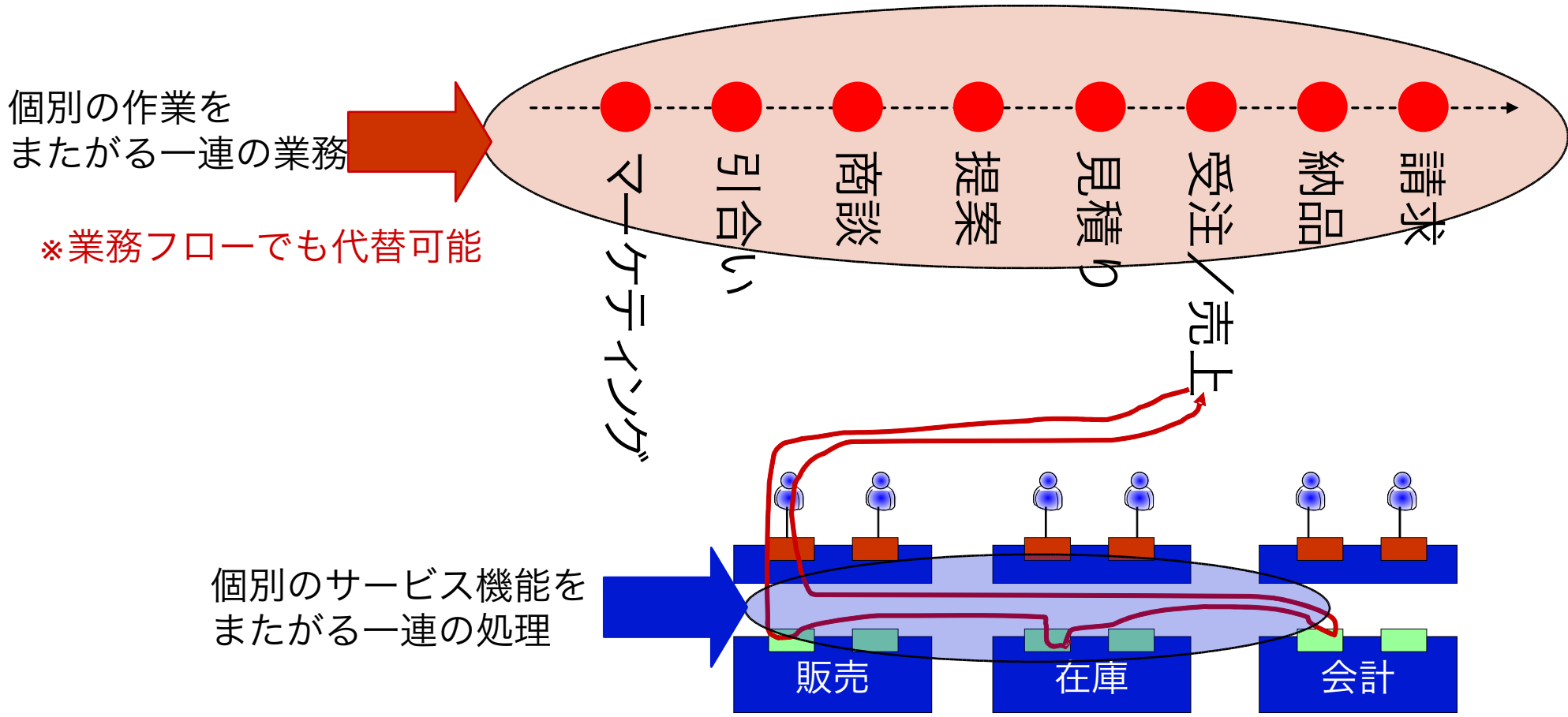
- 業務フロー/ビジネスユースケース



- ビジネス概念モデル (TFP分割法)



ビジネスユースケースを2つのレベルで考える



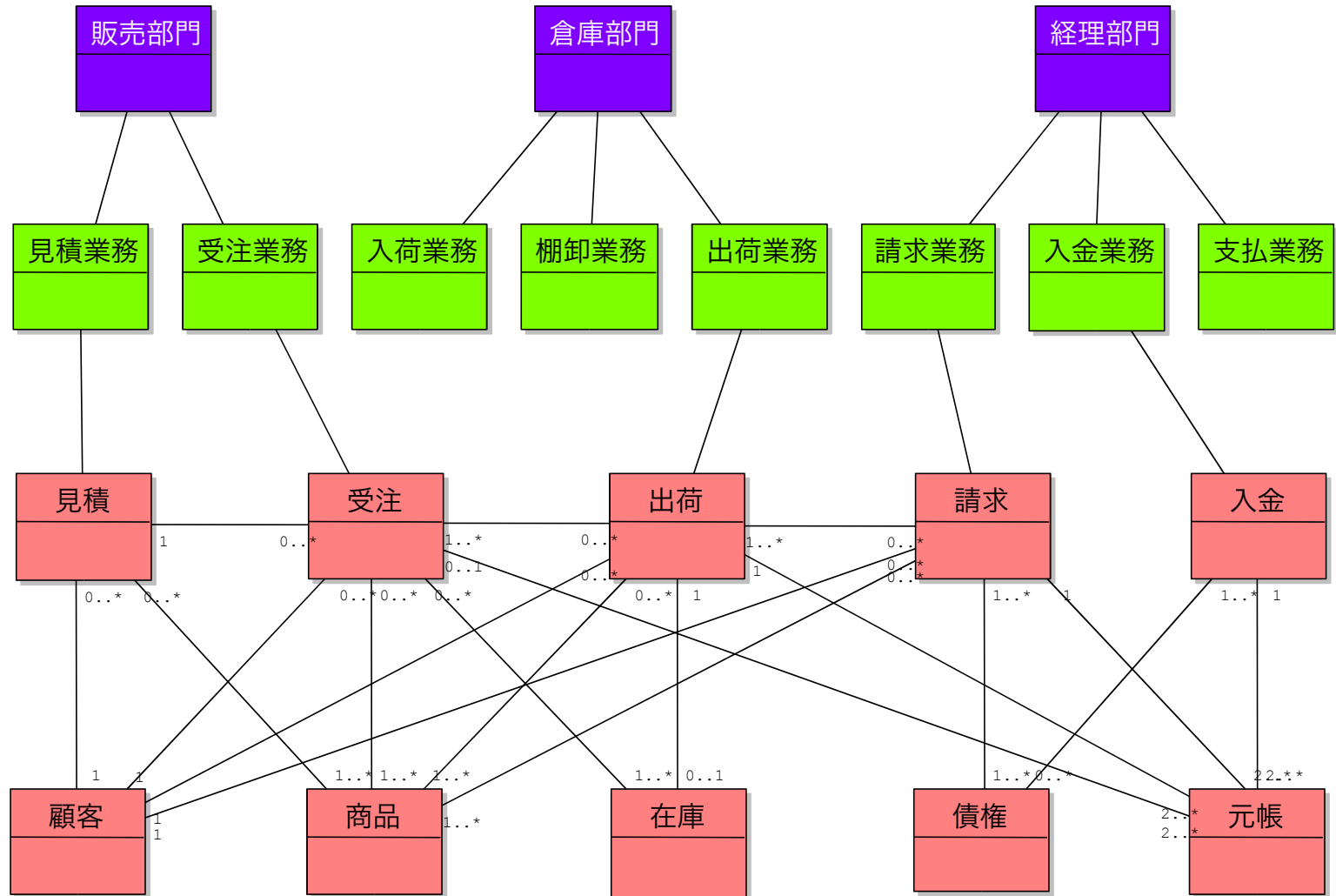
でも、ユースケースの中身はどうする？

(お薦め) TFP分割法

Place

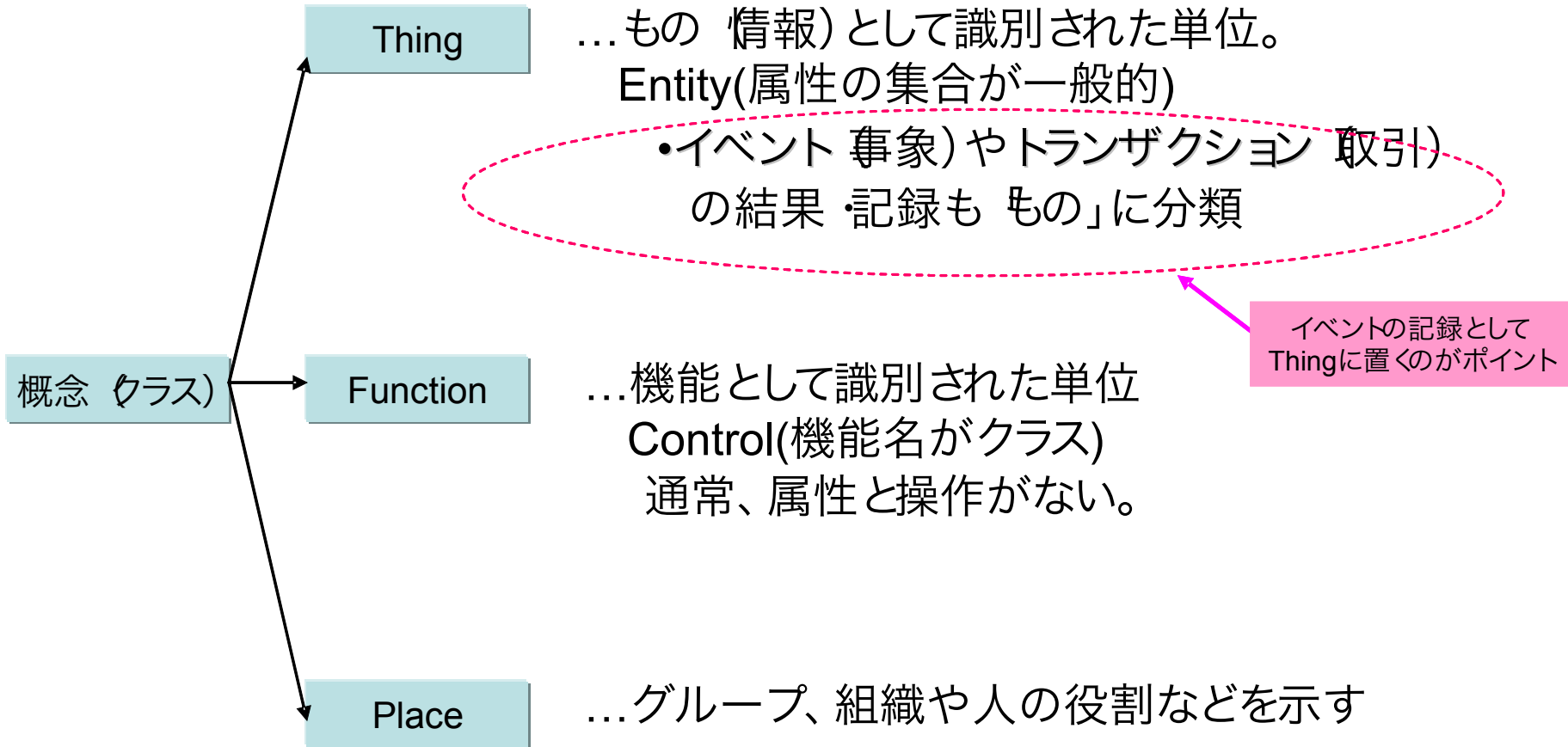
Function

Thing



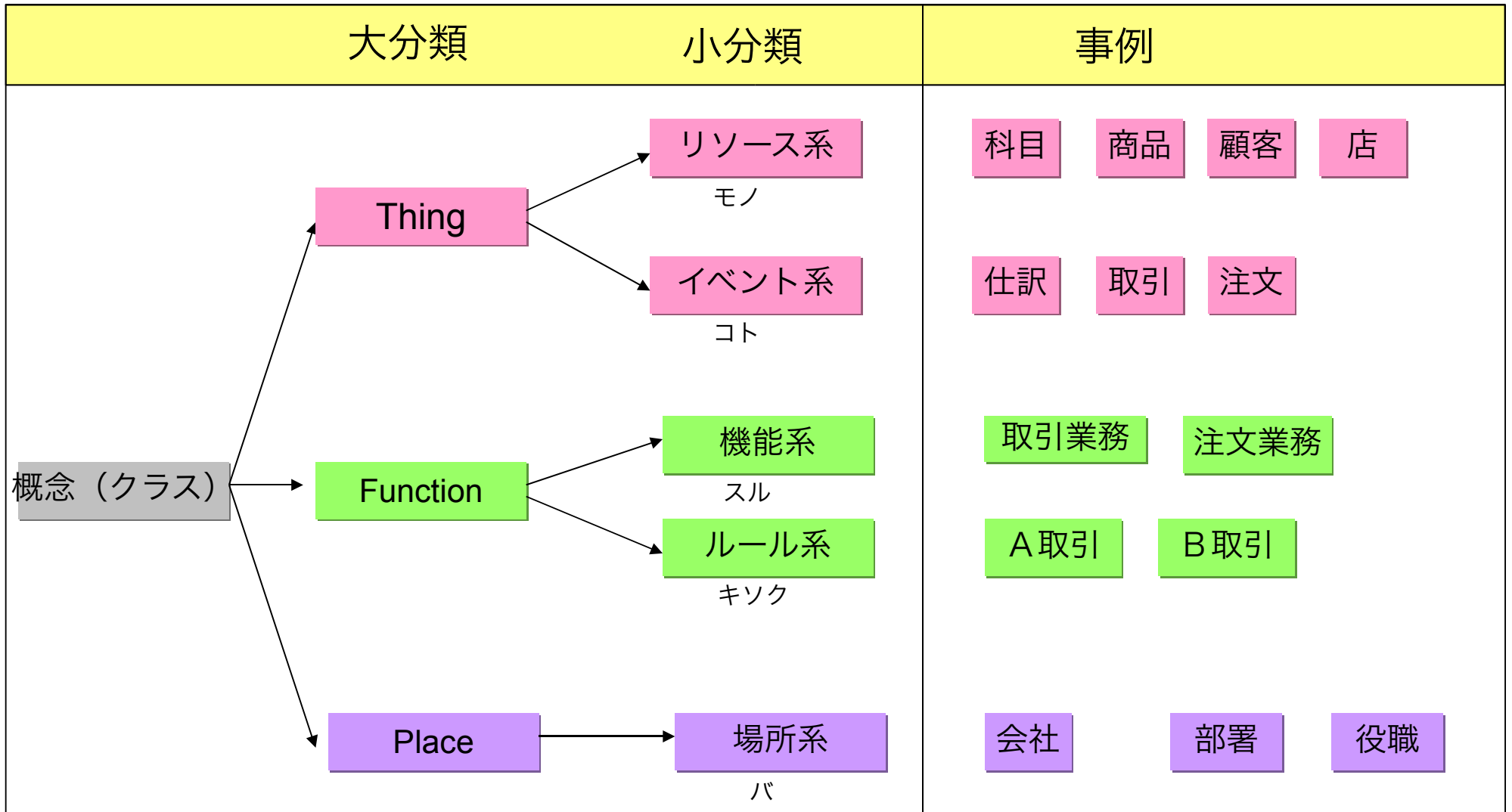
TFP分割法の概要

Thing、Function、Placeにクラスを分類する



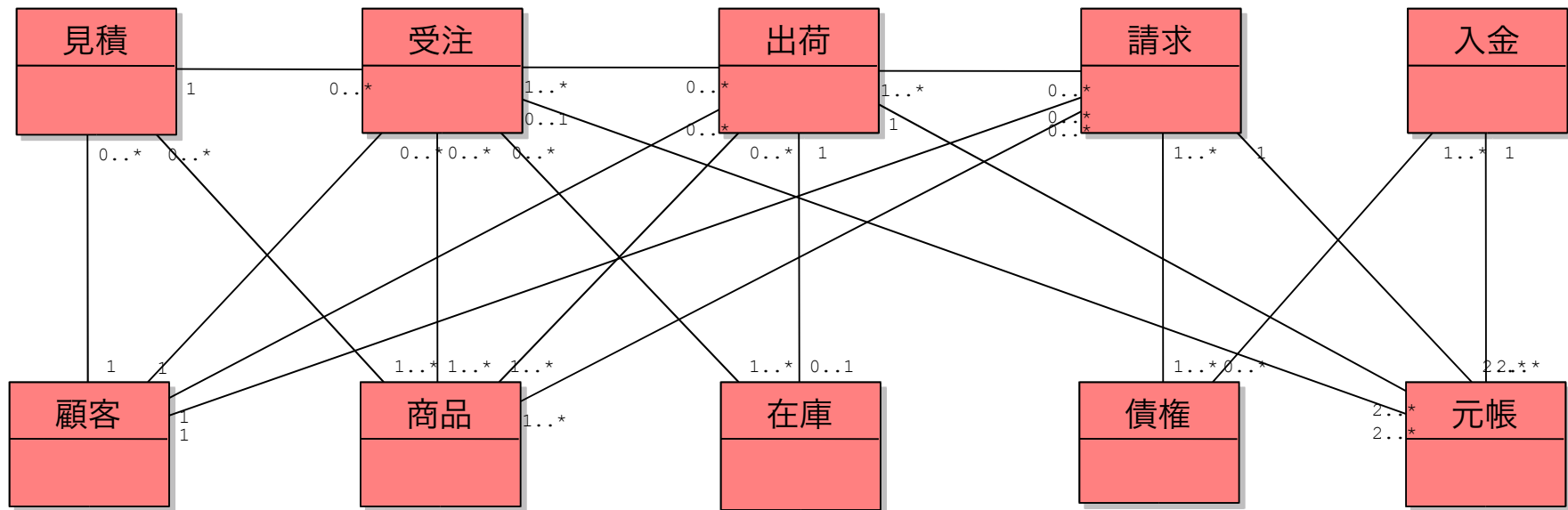
TFP分割法の概要

クラスを Thing、Function、Placeの中でさらに分類する

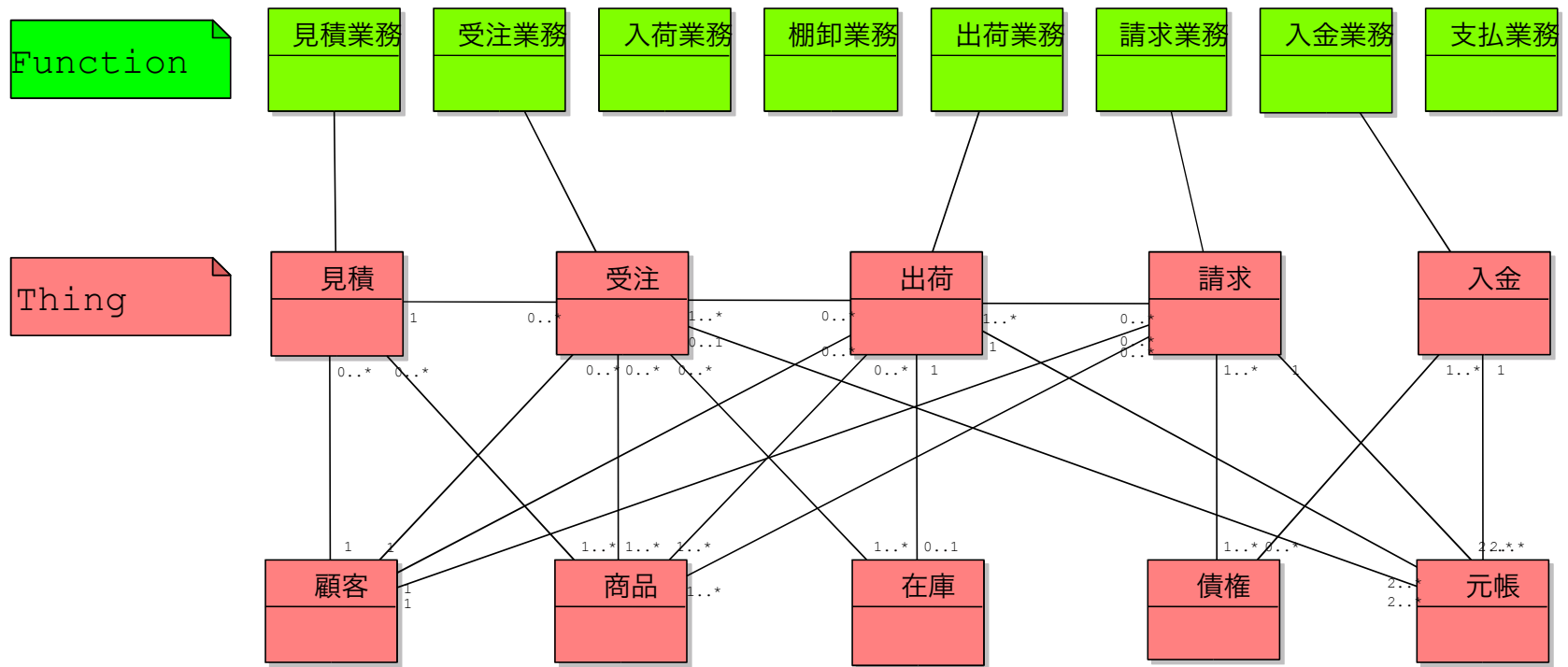


ものともものとの関係

<<Thing>>

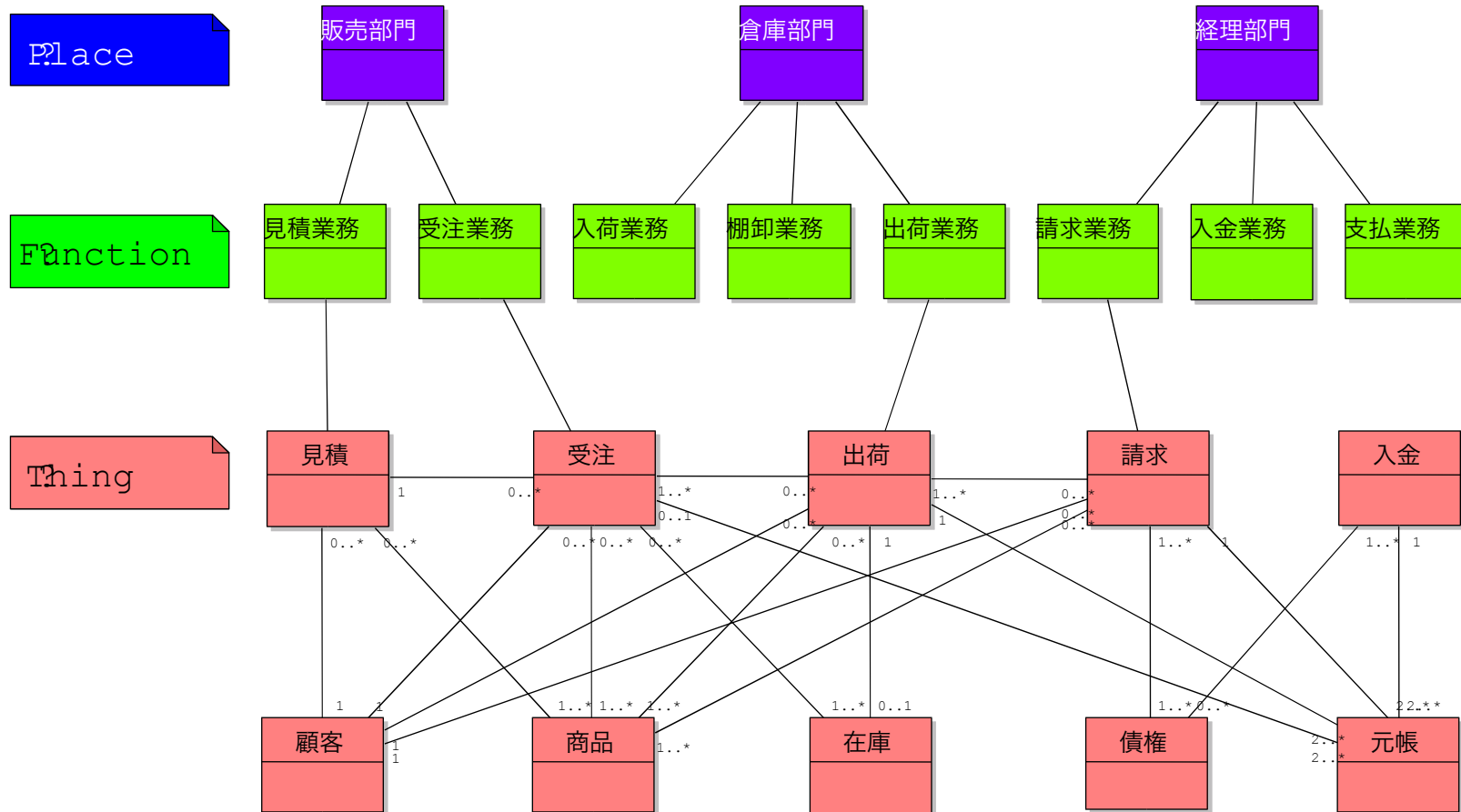


いわゆるドメインモデル



ものの使われ方 (ものの価値証明)

もの・ことを置く場の定義



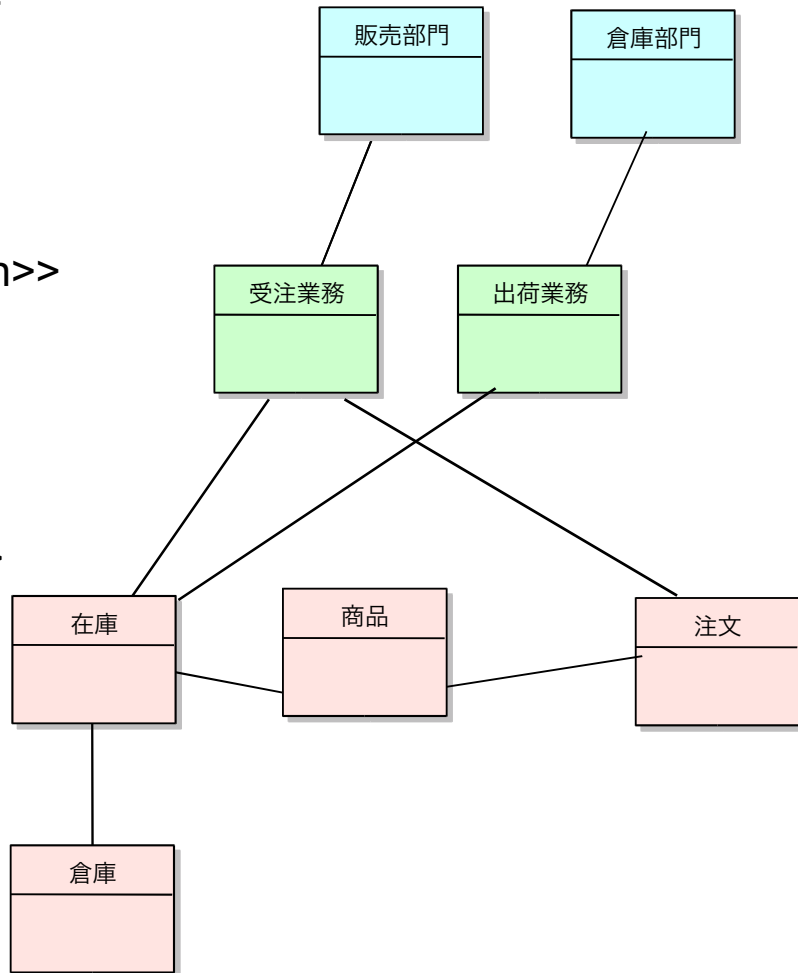
Place図の描き方

どっちの描き方でもOK!

<<Place>>

<<Function>>

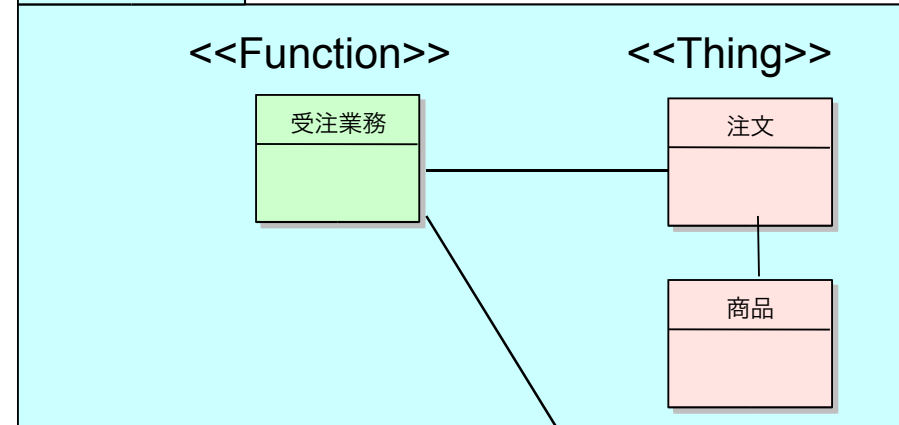
<<Thing>>



販売部門

<<Function>>

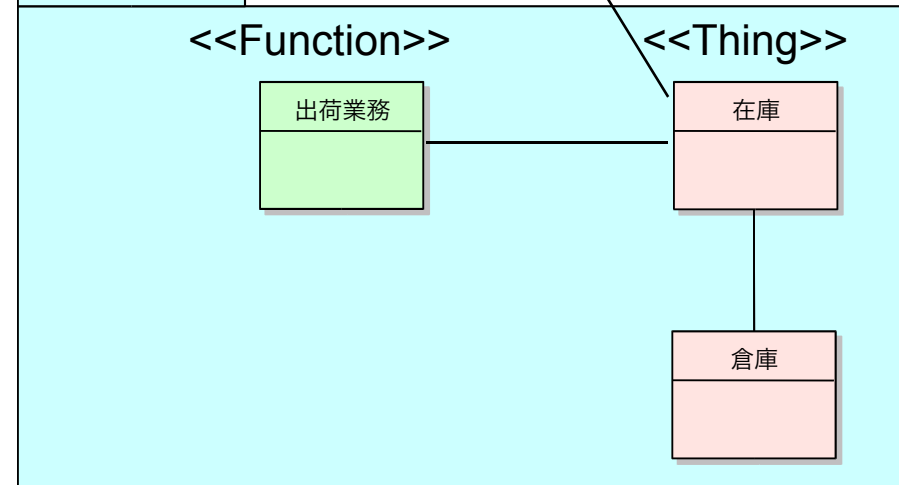
<<Thing>>



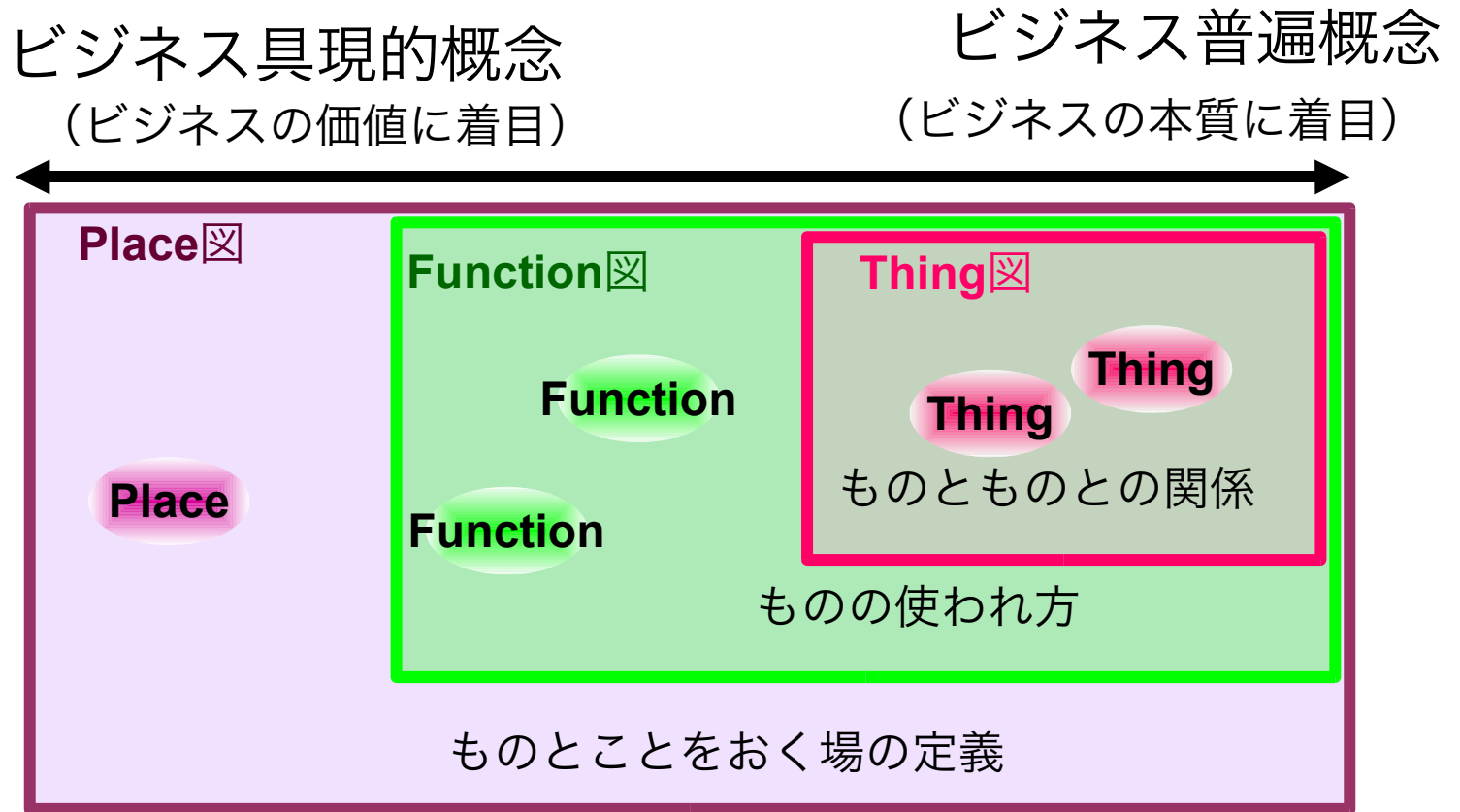
倉庫部門

<<Function>>

<<Thing>>



Thing、Function、Placeで分類する意義とは



分析対象

SOA

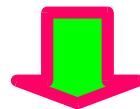
ビジネス
オブジェクト

汎用概念

開発に移行
可能な概念



分散協調設計



アーキテクチャ設計

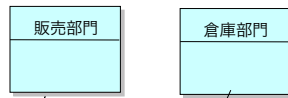


DB設計

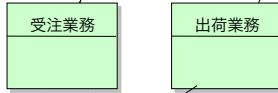
TFPからSOAへ

TFP分析

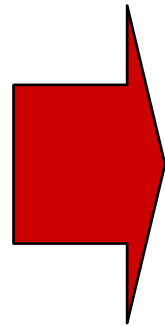
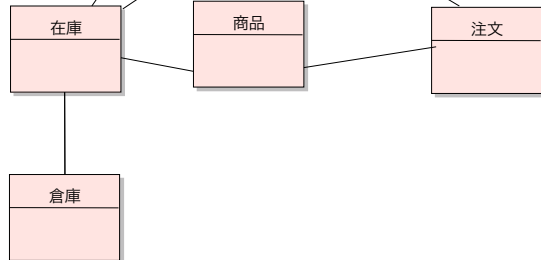
<<Place>>



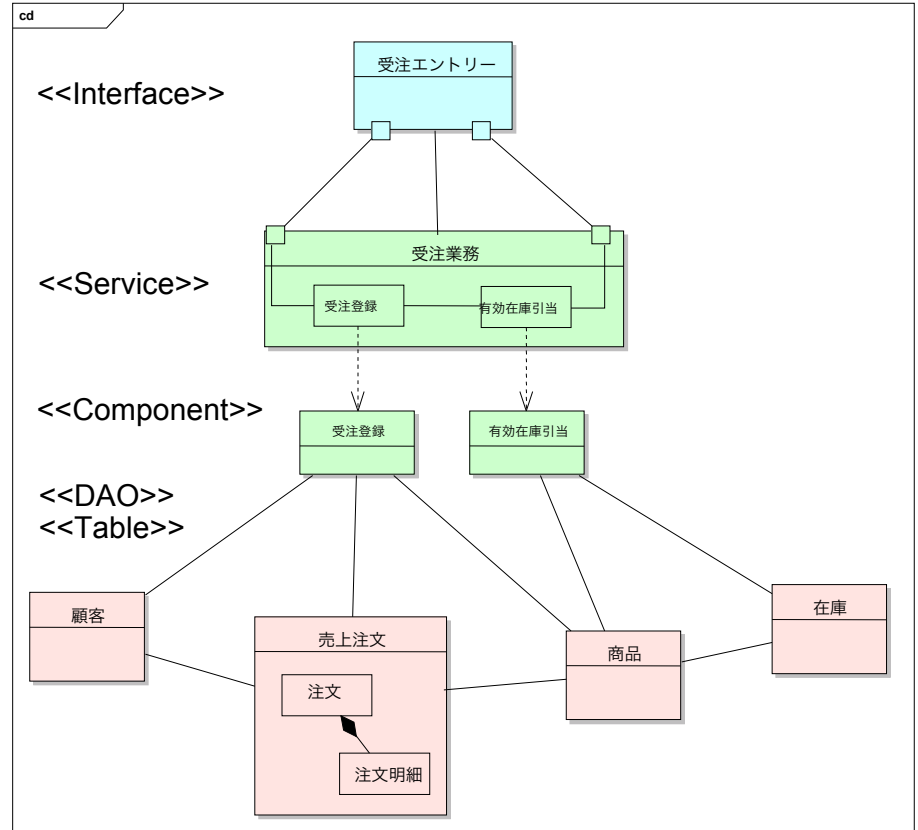
<<Function>>



<<Thing>>

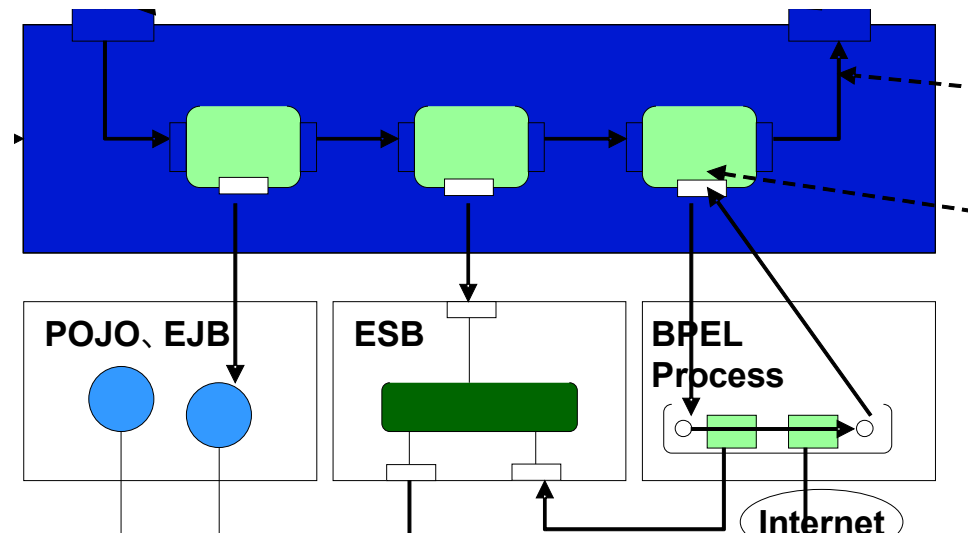
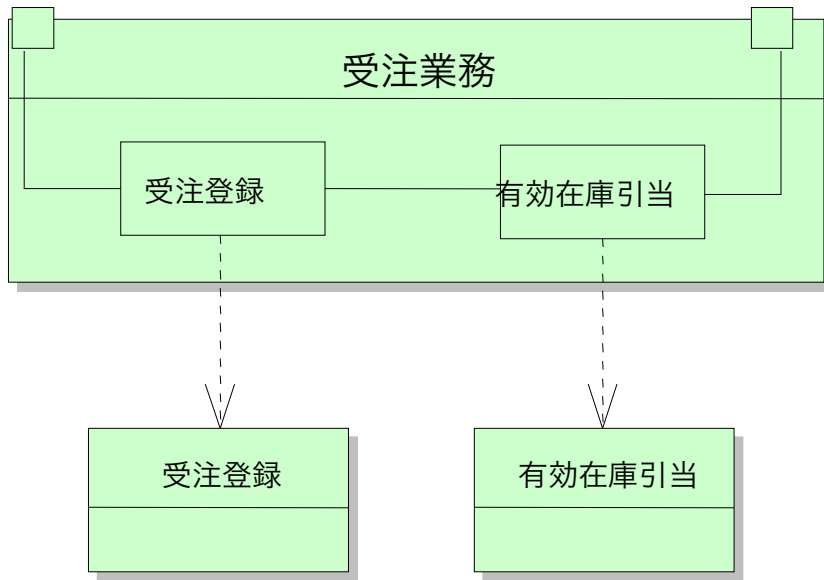


SOA設計



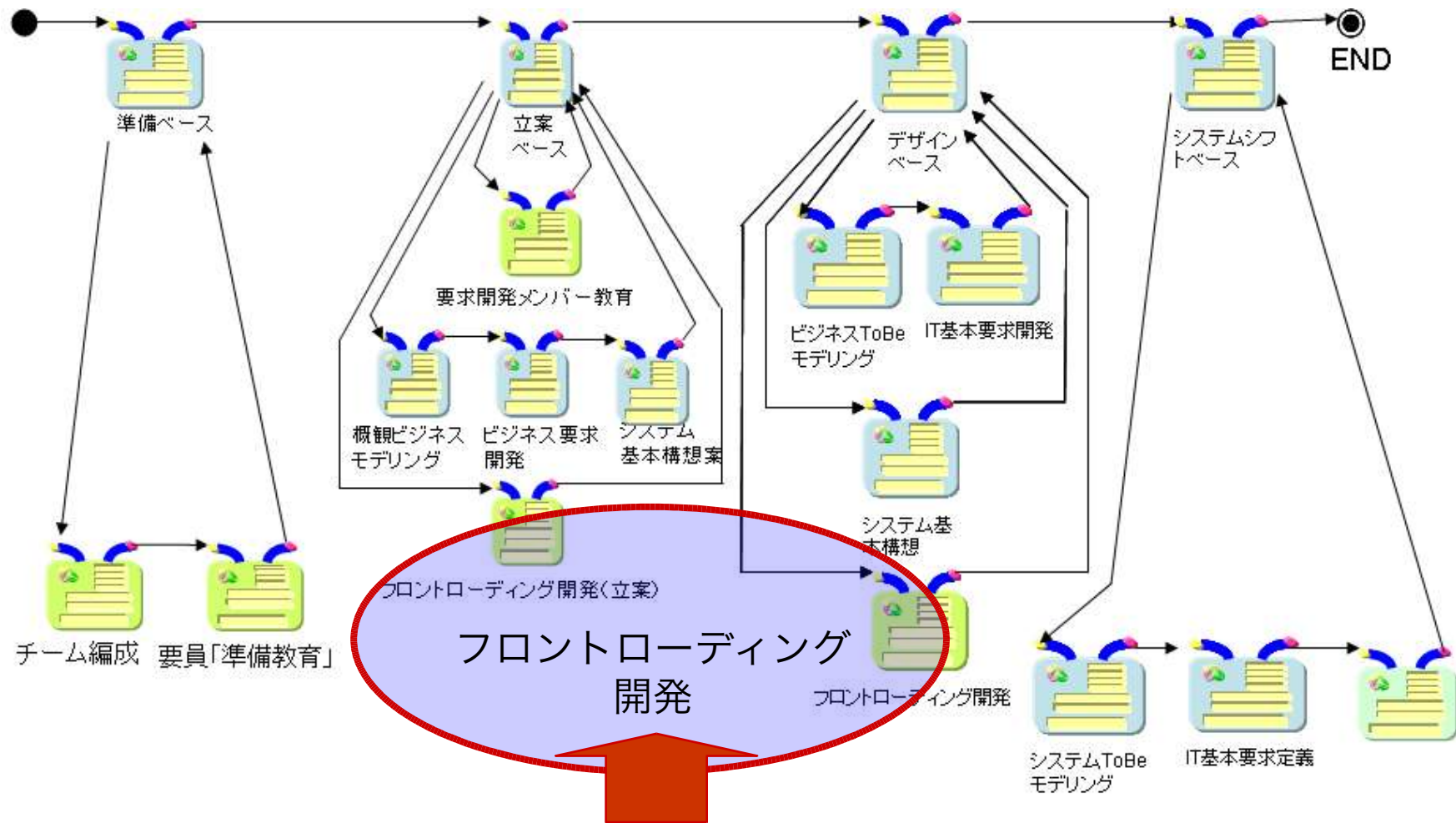
TFPからSOAへ

<<Service>>



<<Component>>

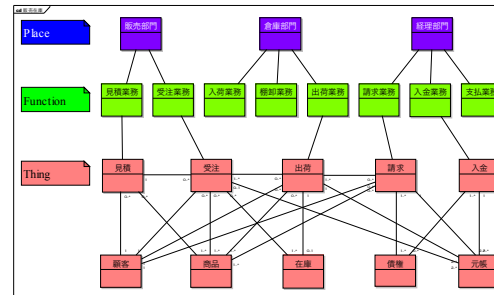
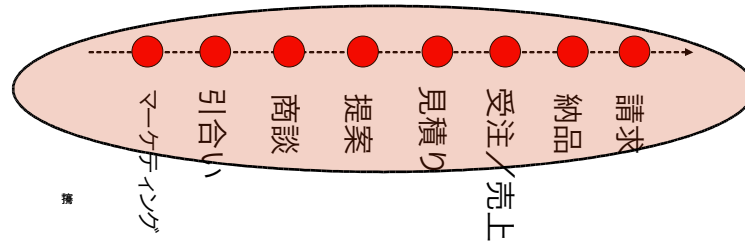
そこでフロントローディングの話です



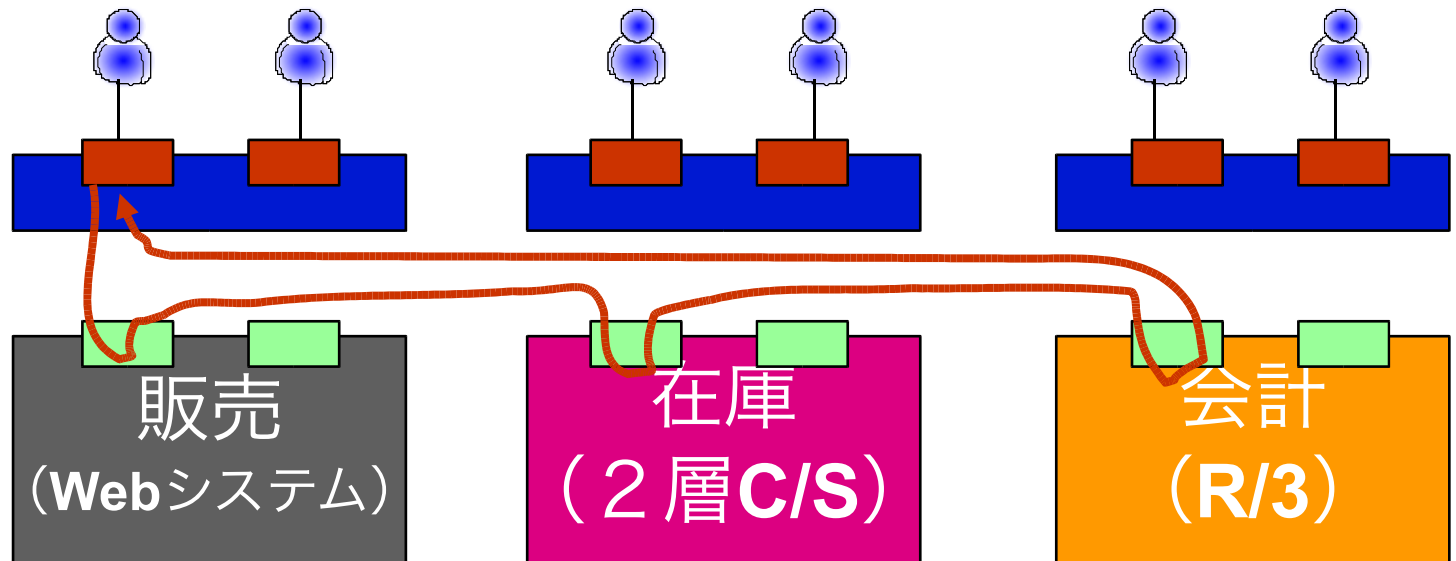
シナリオ①

既存システムを活用した SOAフロントローディング開発

要求開発
モデリング



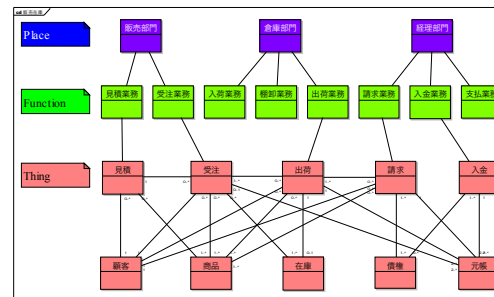
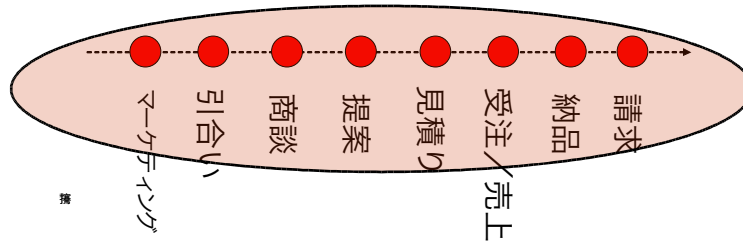
フロント
ローディング
開発



シナリオ②

モックサービス（半完成品の業務サービス機能）を使ったフロントローディング

要求開発
モデリング



フロント
ローディング
開発

