FIT 2014

イノベーションを生み出すビジネスモデルに
直結した情報システムを構築するには

ビジネスモデルから企業情報システムへ

2014年9月4日

情報システム総研
児玉公信

私の立場

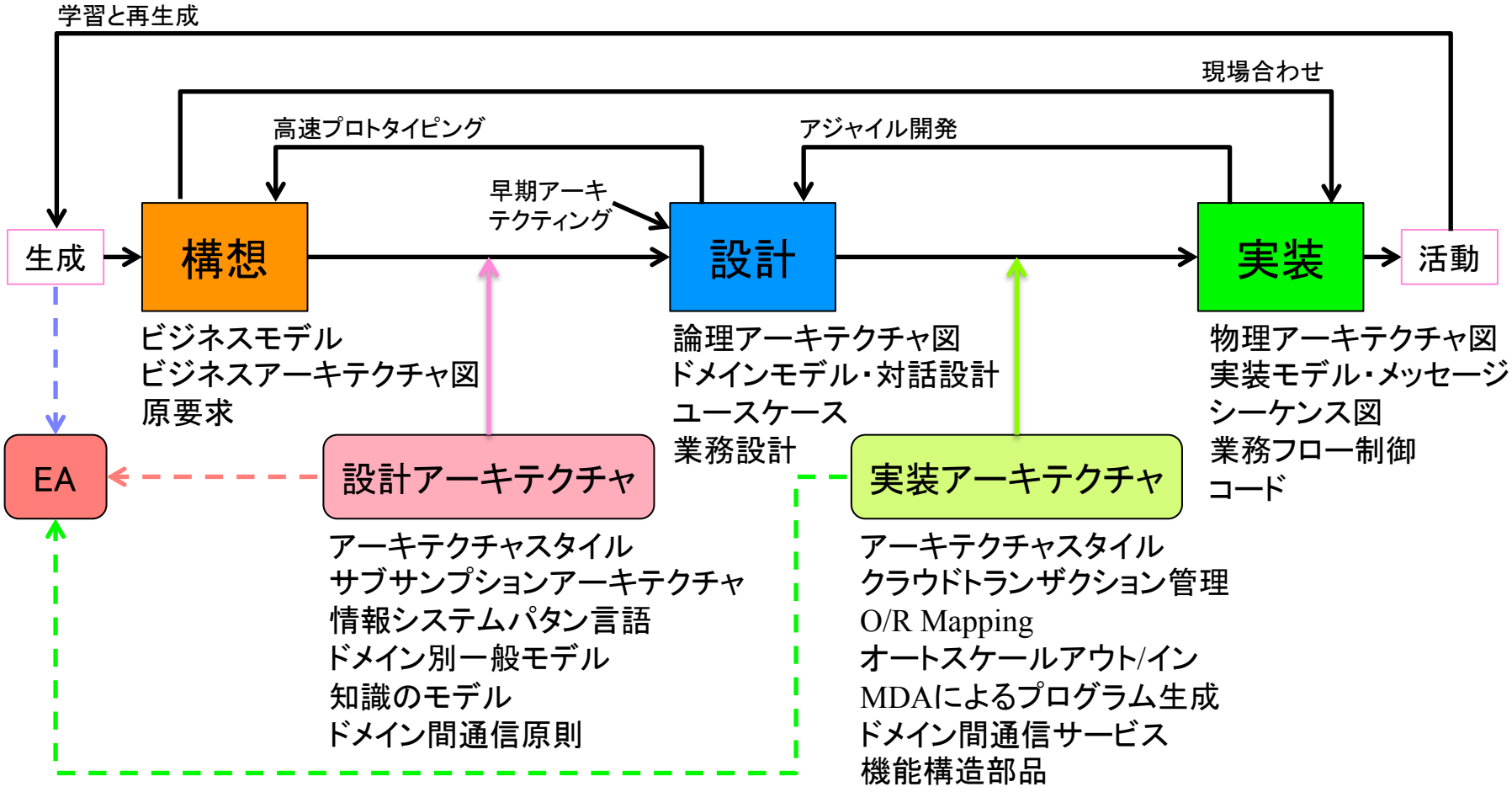
- 情報システム総研 取締役副社長
  - モデラー／アーキテクト
  - 技術士（情報工学）、博士（情報学）
  - 非常勤講師
  - 情報処理学会
    - 技術士委員会委員長、情報処理教育委員会委員、…
    - 「情報システムと社会環境」研究会運営委員
- コンピテンシ
  - 概念モデリング、ドメインモデリング
    - 「UMLモデリング入門」（日経BP）、「UMLモデリングの本質」（日経BP）、「アナリシスパターン」（ピアソン、翻訳）、「リファクタリング」（オーム社、翻訳）、…
  - システムアプローチ
    - ソフトシステム方法論、システム思考（因果ループ図）、ワークデザイン（「もの・こと」分析）、…

システムアプローチ

- システム
  - 多数の構成要素が有機的な秩序を保ち，同一目的に向かって行動するもの（JIS Z8121）
- システム特性
  - 創発（Σ要素の振る舞い＜全体の振る舞い）
  - 通信，コミュニケーション
  - 階層化と機能分担（自己組織化）
- システム思考
  - 「もの」ではなく，目的を持った全体の過程として解釈する
  - 要素に還元できない全体を確認する
    - 全体の中で要素の位置づけを知る
    - 全体が要素を規定すると同時に，要素たちが全体を規定する
    - 原因と思っているものが結果であることもある
  - 分析還元主義に対するアンチテーゼ：全体論的（Holism）

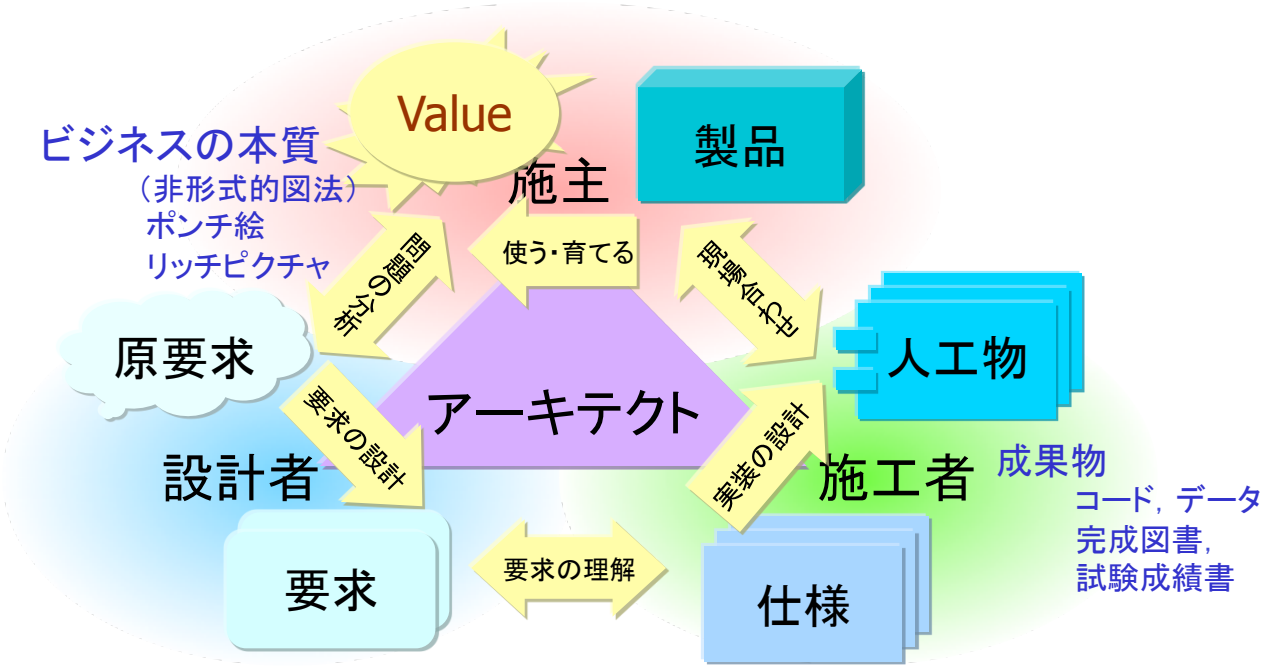
継続的なビジネスモデルの再生成

■ 情報システムサイクルとアーキテクチャ



情報システムサイクル

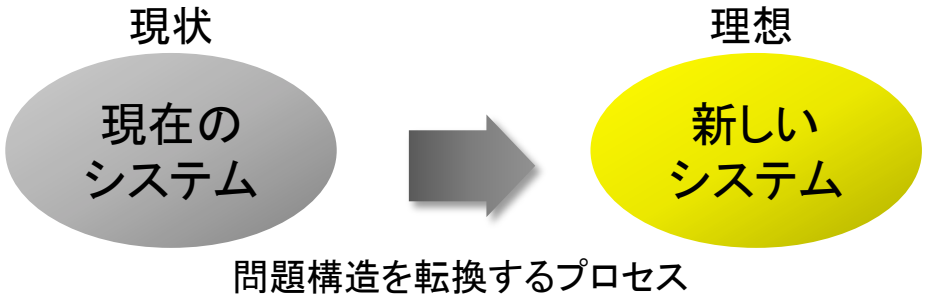
- 真のステークホルダ
  - 施主，設計者，施工者；アーキテクト
    - 役割と責任
    - 繰り返し
    - 合意形成



構想

原因除去で問題は解決しない

- 問題解決におけるシステムアプローチ
  - 原因除去で問題が解決するとは考えない
    - システムは問題も創発する
    - 問題はシステムの弱いところに表面化する
    - システム内の要素は問題状況に抗えない
    - 全体最適を狙う
- 新システムの構想
  - 単なる絵空事でなく，理想を疑似体験する
  - actual に思い描く
  - 施主の「思い」を語る
- 業務シナリオ
  - 業務のかたまりごとに
    - プロット
    - 天地人

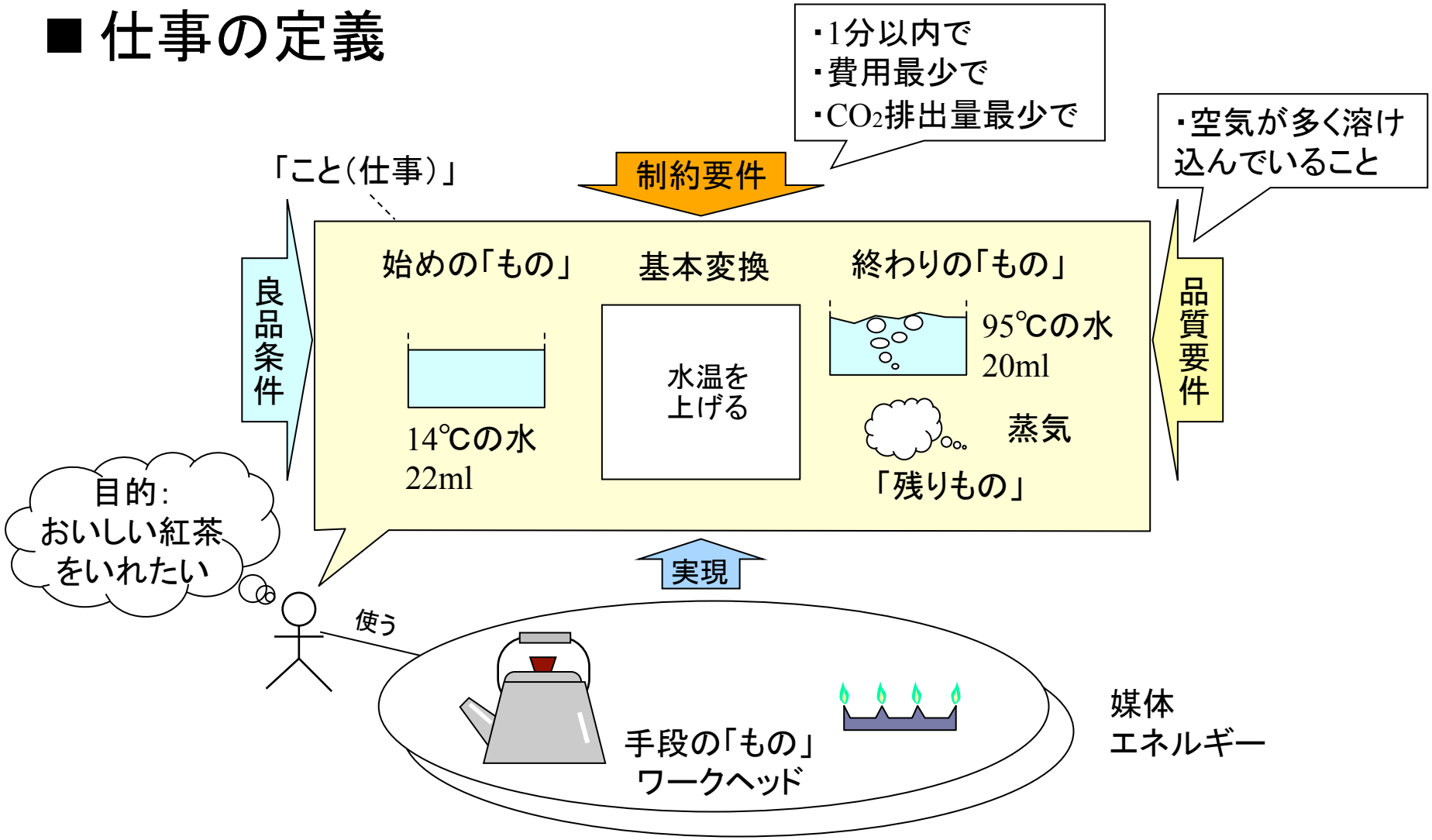


問題構造を転換するプロセス

設計

目的，仕事，制約，手段

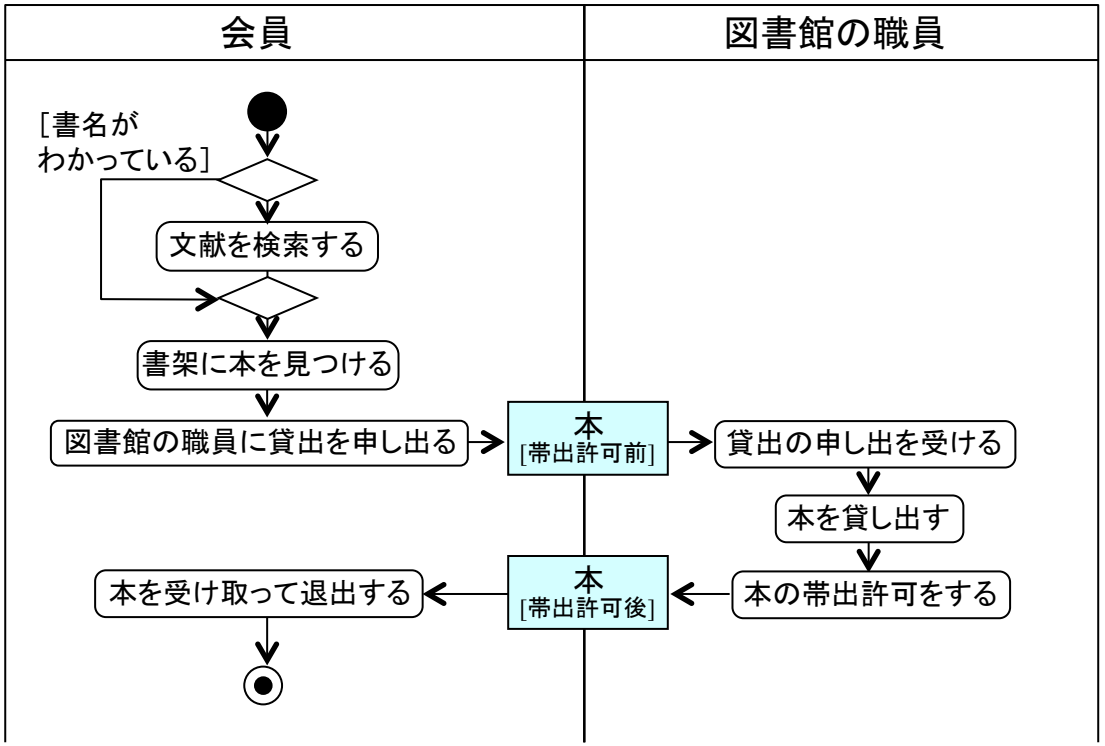
■ 仕事の定義



中村善太郎：「シンプルな仕事の構想法」，日刊工業新聞，1992

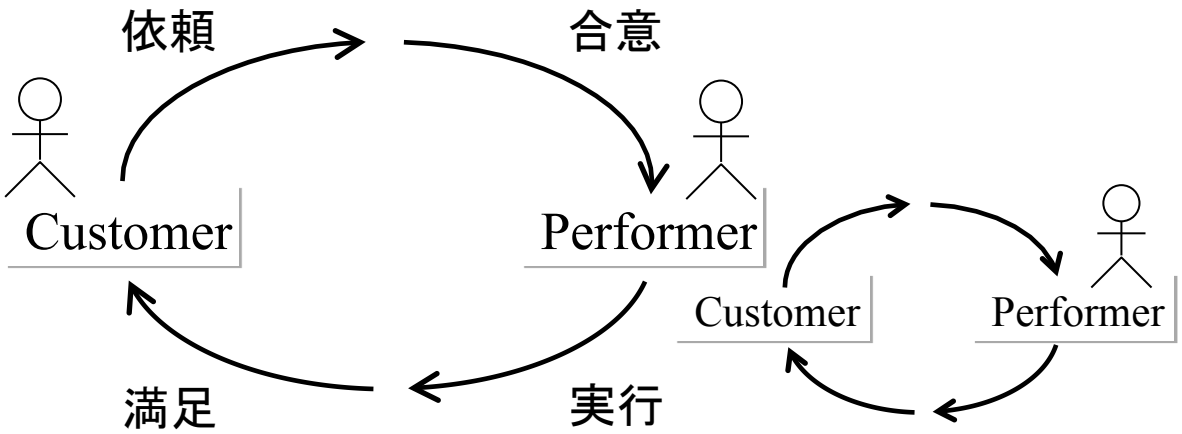
業務フロー図

- 業務＞仕事＞行為の連続
  - アクタ間の依頼–受託関係
  - 機能の流れにしない
  - *Customer-Performer*の関係



Customer-Performer

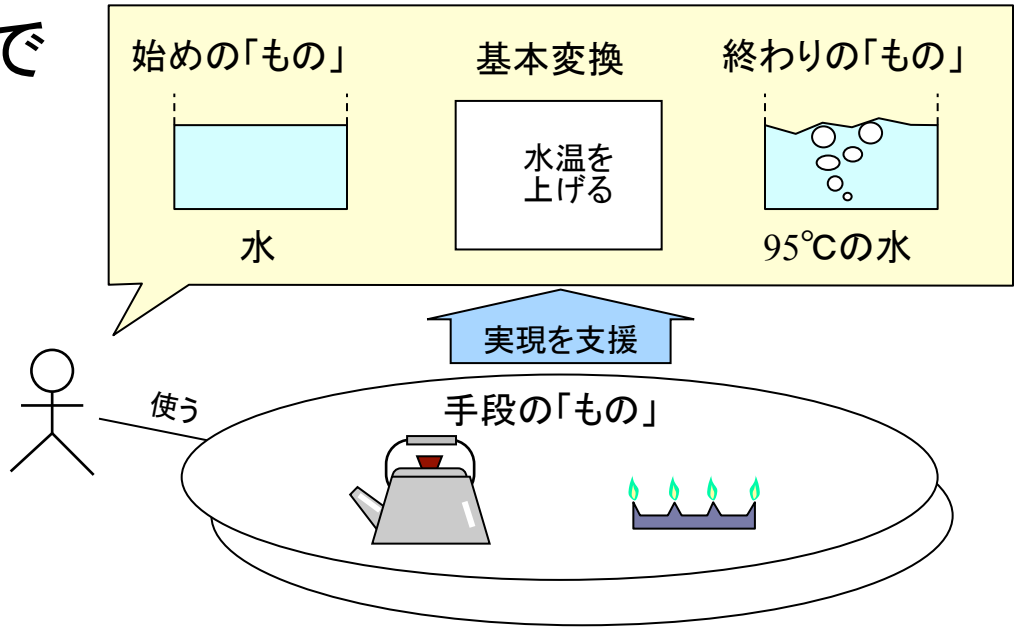
- 業務フローの本質
  - 対話の4フェーズ
    - 提案→合意→実行→満足
    - 分割と下請け
    - フローの*starter*に戻る



Medina-Mora, Winograd, Flores, and Flores, "The Action Workflow Approach to Workflow Management Technology," Proc. of CSCW 92, pp1-10, 1992

機能の定義

- 機能とは
  - 「始めのもの」を「終わりのもの」に変換する操作
  - *Function*（関数=機能）
- 機能と手段を分離する
  - 手段を選択する前に，機能とその制約を明らかにする
  - 手段は施工者と検討
- ユースケースの形式で

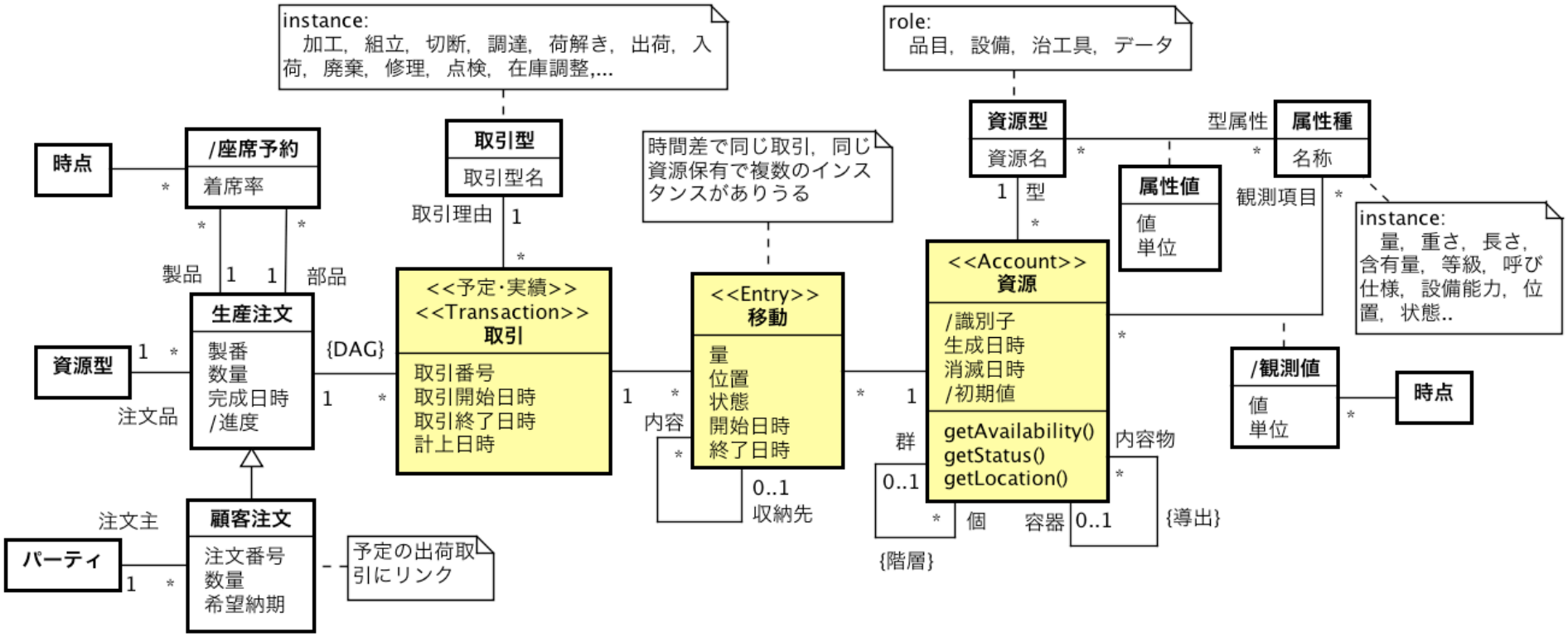


アーキテクティング

ドメインの一般モデル

■ 生産管理ドメインの一般モデル（CHARM3）

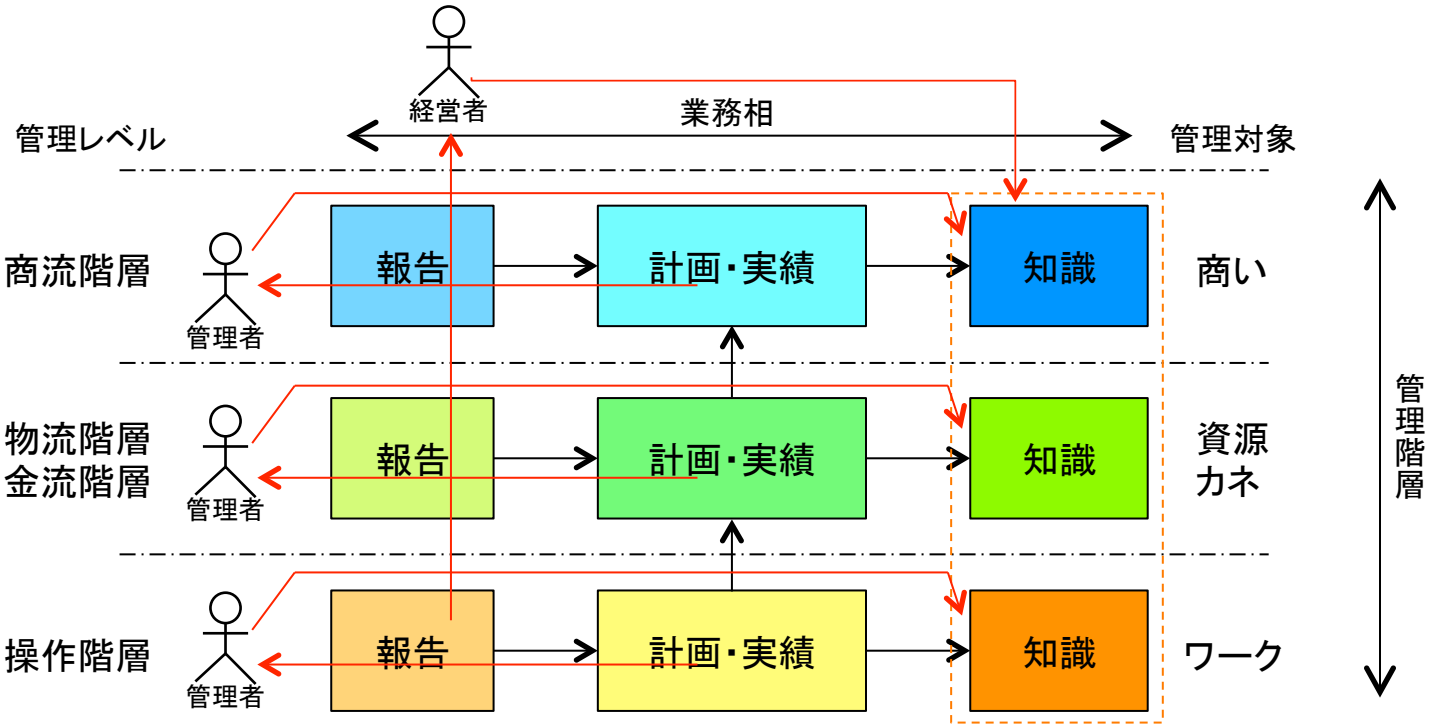
Cross Hierarchical Account=Resource Model v.3



児玉公信，水野忠則，「少量多品種型生産管理システムの一般モデルCHARMの提案」，
情報処理学会論文誌，Vol. 49(2), 902-909, 2008

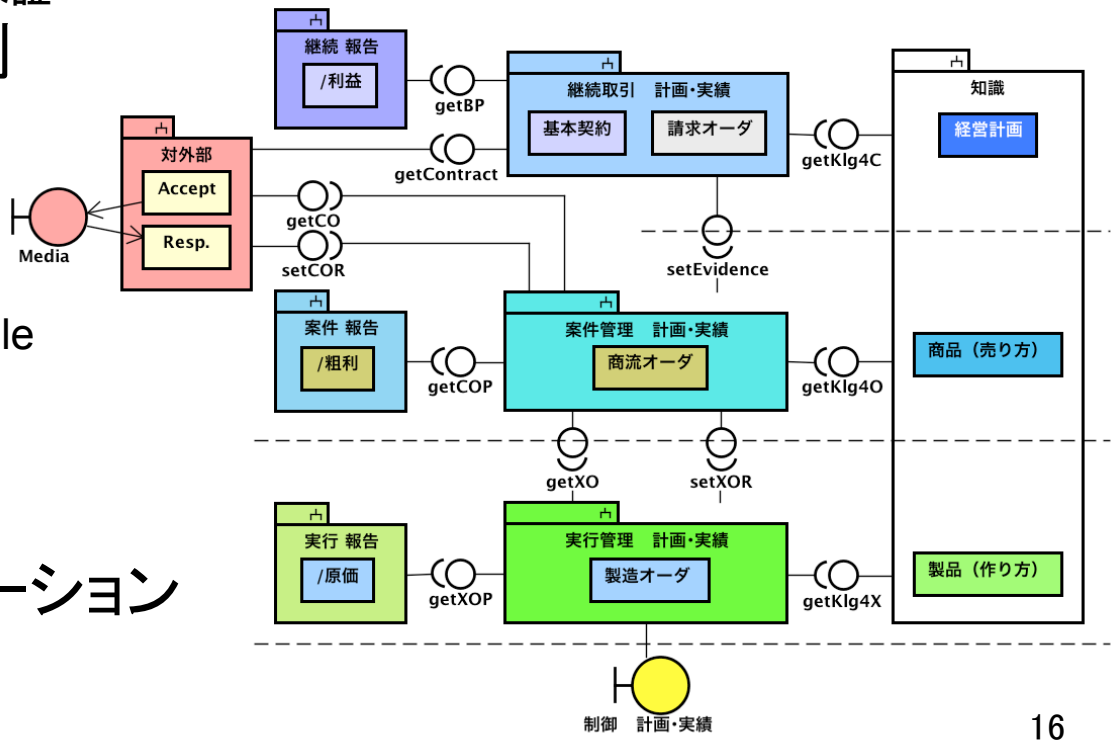
システム分割の指針

- 管理階層と業務相に基づく分割
  - 管理階層ごとに、ライフサイクルを持つ関心対象
  - 業務相は，PDCAサイクル
    - 知識に基づく計画展開
    - 報告相での評価と知識の調整



論理アーキテクチャ設計

- システム間連携モデルに沿ってモジュールを設定
  - ドメインモデルの実装とユースケース実現
    - ドメイン内は機能強度（cohesion）を高く→トランザクション境界
    - ドメイン間は低い結合度（coupling）で，疎に連携
      *semantics*を共有しないメッセージ（タグ付き）
      非同期通信，冪等性の保証
  - インタフェースの原則
    - 下から上へ一方向
      指示系
        上からNotify
        下からgetRequest
        Observer-Observable
      実績系
        下からsetState
- 外部複雑性の分離
  - 契約に基づくバリエーション



アーキテクチャ以前の問題

組織の Cultivationと人材の育成

- 組織のアーキテクチャ成熟度（AML）
  - 組織が背負っている歴史と風土
    - 組織のアーキテクチャ成熟度以上のことはできない
    - アーキテクチャ成熟度を上げるためには，組織と人を作り直す
- 人の作り直し
  - システム思考の導入
    - 視点の注入
  - プロセス進度に応じた教育
    - 丁稚奉公
    - 基幹システムのモデリングとアーキテクティング
    - 早期プロトタイプの作成
    - 発注訓練
- 持続的な改善ができる風土づくり

真の施主をあぶり出す

- ほんものの施主
  - 抽象思考：現実からの切り離しと，中長期展望
  - 作り方の作り方，要求定義の要求定義→ガバナンス
  - ほんものの施主が発見されるまで時間がかかる
    - プロトタイプ開発で，小さく失敗してみる
- 子飼いのチーム
  - 真の施主との直接チャネル
  - 改革のコアチームの変遷
    - 初期段階：CIOチーム
    - 設計段階：アーキテクチャチーム
    - 量産段階：仕様決定チーム
  - 段階に応じて教育