OPENTHOLOGY

Openthologyモデリング ガイドライン

Openthologyプラグイン

Ver 0.6

2005年1月25日

OPENTHOLOGY

モデリングガイドライン ＜＜UMLビジネスモデリング＞＞

OPENTHOLOGY

ビジネスフロー編

（参考）未完成

OPENTHOLOGY

プロセスモデル（業務フロー）

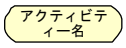
- 表記上の制約条件
  - AsIs,ToBe,Realizeモデル共通
    - アクティビティ（業務処理）の単位が関係者間で認知されていること。
  - AsIs,Realizeモデル
    - システムとの関係が明確になっていること
      - システムレーンを設けるか、アクティビティで識別表現できる事
    - システムの入出力が明確であること
      - できるだけ業務担当に理解しやすいメタファ（帳票・画面）を利用
  - ToBeモデル
    - 概念モデルとの関係が明確になっていること
      - ToBeモデルでは、画面・帳票などは抽象化された概念オブジェクトとして表現される
        - » 概念オブジェクトの妥当性がモデル上で検証できること

4

OPENTHOLOGY

まずはUMLアクティビティ図から

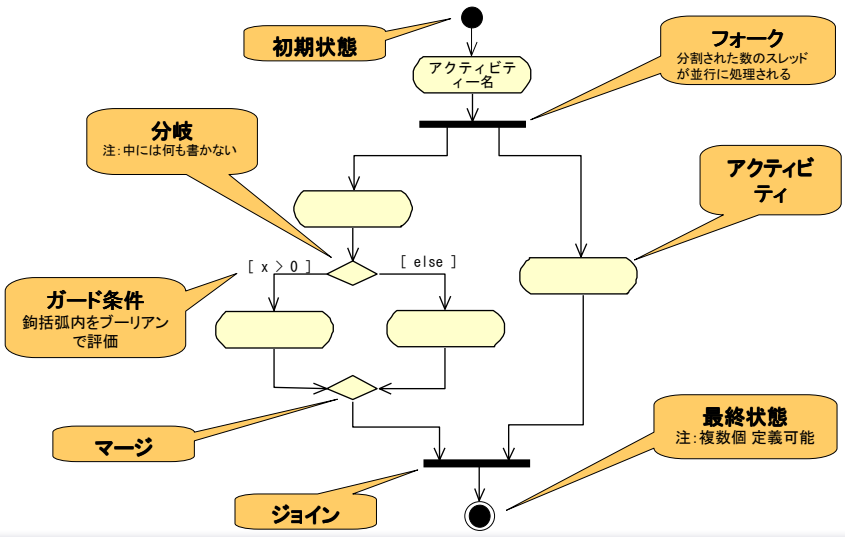
- まずはUMLのアクティビティ図を使って業務の流れとシステムの関係を捉えます。
- アクティビティ図とは、業務や処理のフローを記述するための図のことです。
- アクティビティとは1個以上の業務処理を意味のある単位にグルーピングしたもので下記のように記述します。



5

OPENTHOLOGY

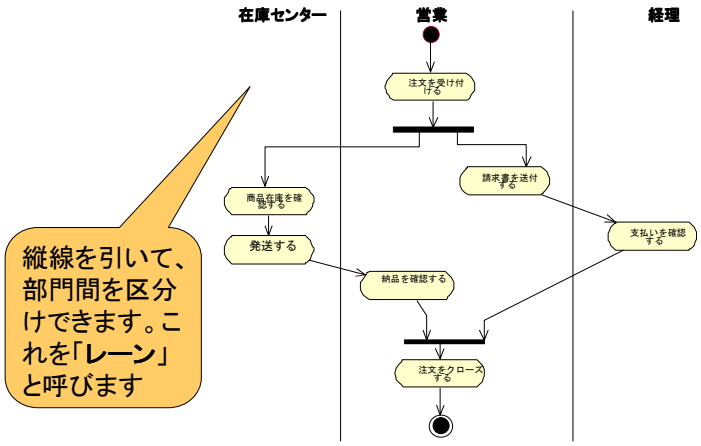
UMLアクティビティ図



6

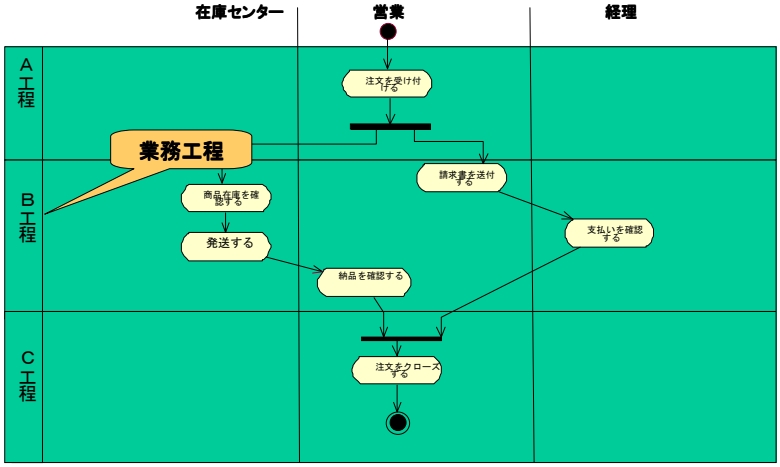
OPENTHOLOGY

UMLアクティビティ図の例



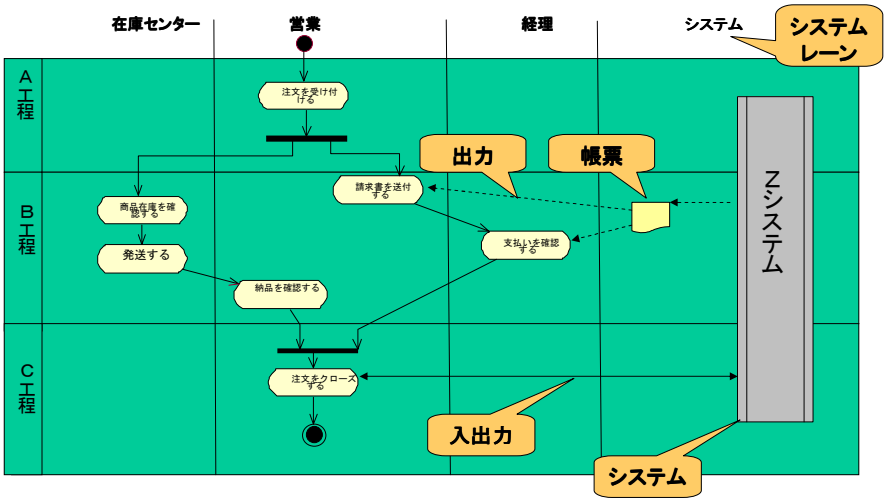
OPENTHOLOGY

業務プロジェクトに向けて追加した表記部分



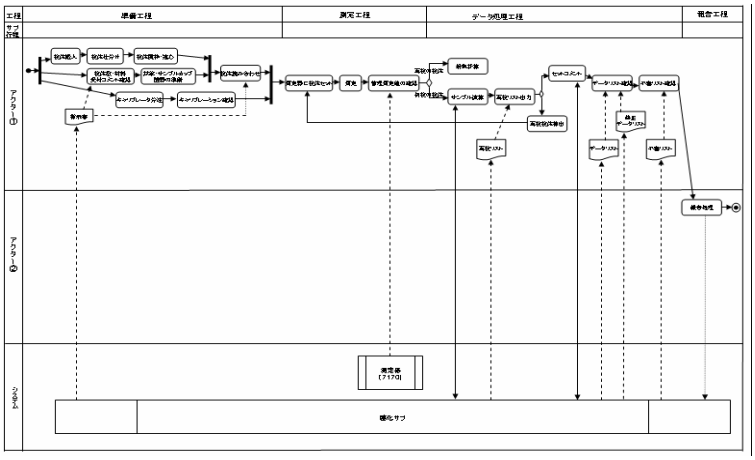
OPENTHOLOGY

業務プロジェクトに向けて改善した表記部分



OPENTHOLOGY

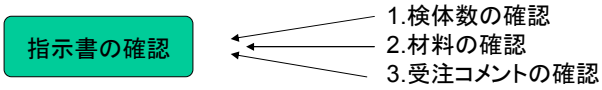
業務モデリング例



OPENTHOLOGY

アクティビティの粒度

- アクティビティという単位は次のような基準で決まります。
  - 1個以上の作業から構成される意味のある一塊の作業まとまり
  - アクティビティは、業務処理を意味のある単位でグルーピングしたもの
    - 例:「指示書の確認」というアクティビティは、下記をまとめたものです。下記は指示書中に確認すべき情報がまとめられており、作業的に切り離しできません。
      - 検体数の確認
      - 材料の確認
      - 受付コメントの確認
    - 基準:新人に自分の業務を指導する際に分かりやすい、業務概念の単位
  - この時、「1. 検体数の確認」もアクティビティと考えることができますが、このレベルでは、作業を理解する上で細かすぎるという問題があります。

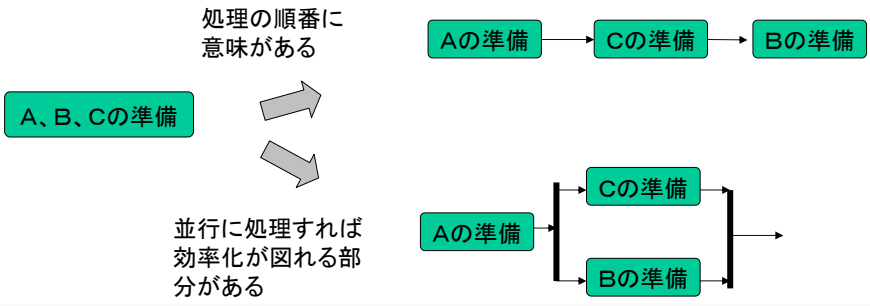


11

OPENTHOLOGY

アクティビティ(粒度の基準)

- 同じ業務上の役割(作業者が一人)で行える単位
  - 例:「A、B、Cの準備」は、3つに分かれるかもしれない。
- 同じタイミング(ライフサイクル)で行う業務単位
  - 異なるタイミングで行うものは一つのアクティビティにしない。
  - 並行的に進められる作業を一つのアクティビティにしない。



12

OPENTHOLOGY

ステレオタイプ

- ステレオタイプは、アクティビティのスーパークラスを示します。スーパークラスを示すことで、個々のアクティビティを抽象化することができ、それが理解を深めるものであれば、ステレオタイプをつけます。

（例）

<<機械処理>>
マニュアルモード

<<機械処理>>
オートモード

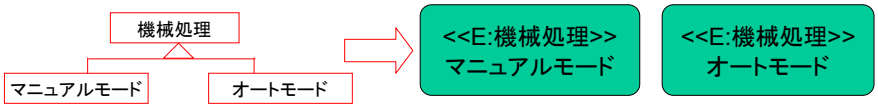
13

OPENTHOLOGY

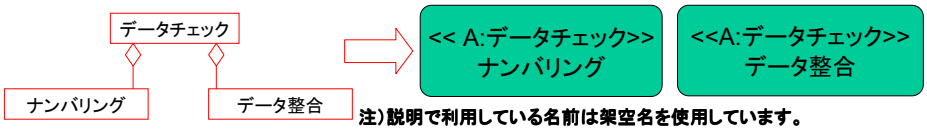
ステレオタイプの詳細化

- ステレオタイプは、アクティビティを抽象表現したものと書きましたが、実際には、集約関係と汎化関係の2つのタイプがあります。現段階では、この2つのタイプを差別化していませんが、今後差別化する際には、下記のように行います。

汎化関係の例（マニュアルモード、オートモードは機械処理の一種である）



集約関係の例（ナンバリングとデータ整合はデータチェックの部分処理である）



注）説明で利用している名前は架空名を使用しています。

14

OPENTHOLOGY

アクティビティの標準化

- アクティビティ図が複数ある場合、ある程度アクティビティ図が出てきた段階で、アクティビティの標準化を行います。その後、標準化されたアクティビティによってアクティビティ図を作成するようにします。
  - **アクティビティ名の調整**
    - 同じ処理グループを異なるアクティビティ名で表現している場合、もっとも適切で分りやすいアクティビティ名の方を標準名とします。
  - **粒度の調整**
    - 業務ごとに異なる粒度でアクティビティを定義している場合もあります。その場合、どちらかの粒度でアクティビティ図を表現するのか決定します。ただし、業務によっては、他業務よりも詳細なアクティビティを利用する必要があることもあります。その場合は、細かな粒度での表現も認める必要があります。

OPENTHOLOGY

標準アクティビティ・リスト

- 標準アクティビティ・リストにより、アクティビティの標準化と粒度の説明を行います。

工程	スーパータイプ	アクティビティ名	サブアクティビティ名	利用部門			説明
X工程	E:機械処理	オートモード					オートモードの説明を書きます。
X工程	E:機械処理	マニュアルモード					マニュアルモードの説明を書きます。
X工程		A:データ整合	整合前処理				整合前処理の説明を書きます。

アクティビティが詳細化しなければならない業務のみ、これがアクティビティ名となります。
この場合、アクティビティ名がステレオタイプとして使用されます。((通常省略))
アクティビティの最小粒度がサブアクティビティとなります。

アクティビティ名が記載されます。

スーパータイプは分りやすい場合のみ付与されます。
記述形式 「<<E:xxxxx>>」or「<<A:xxxxxx>>」
((通常省略))

工程名を記述します。

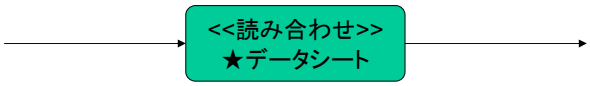
<<A:データ整合>>
S:整合前処理

注)説明で利用している名前は架空名を使用しています。

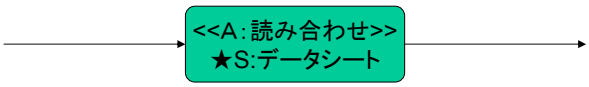
OPENTHOLOGY

特殊なアクティビティ

- ２人で実行されるアクティビティ
  - ★印を付与します



  - サブアクティビティの場合

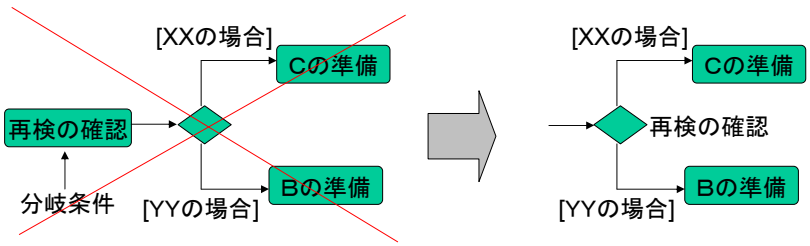


17

OPENTHOLOGY

分岐

- 分岐は、アクティビティの特殊系（分岐処理をおこなうアクティビティ）と考えます。
  - 分岐条件名がアクティビティとして重要な場合
    - 分岐条件名を分岐マークの横に書いてよい
    - 重要でない場合は、省略可

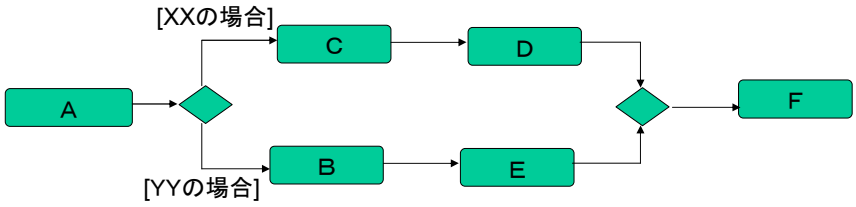


18

OPENTHOLOGY

マージ

- マージは、1つの分岐から流れてきた線を元に戻すために使用します。

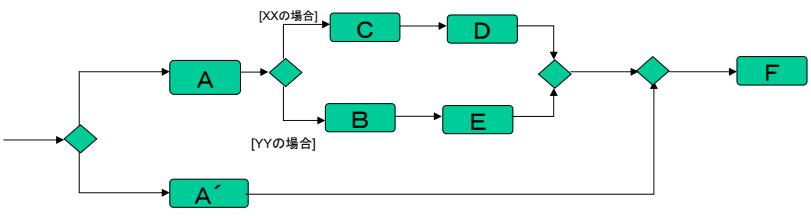


19

OPENTHOLOGY

マージ

- マージは、1つの分岐から流れてきた線を元に戻すために使用します。

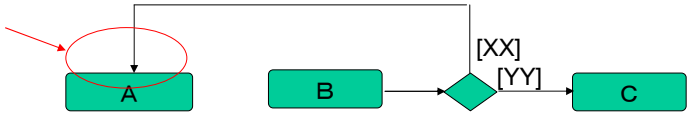


20

OPENTHOLOGY

マージ

- マージを使用しないケース
  - 下記の場合はわかり辛くなるためマージを省略します。
    - 戻り処理について



21

OPENTHOLOGY

ハイレベル表記

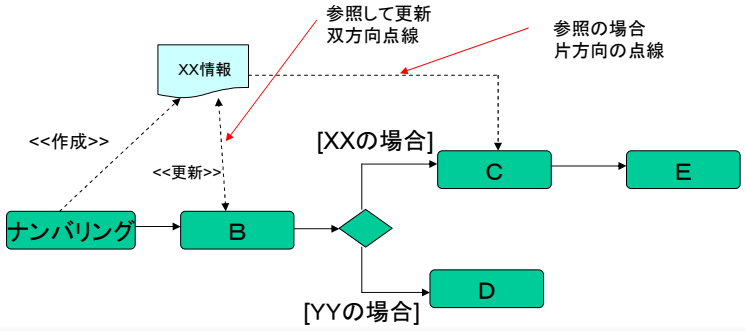
- ここまでの表記法では、どうしても業務を正確に表すことができない場合は、次から説明するハイレベル表記を使用してください。

22

OPENTHOLOGY

ハイレベル　手動による情報の入出力

- アクティビティの結果を情報として記録しておき、後で他のアクティビティから利用する場合。
- システムと無関係な記録を残す場合は、帳票マークに向かう線にステレオタイプ<<作成>>を使用します。

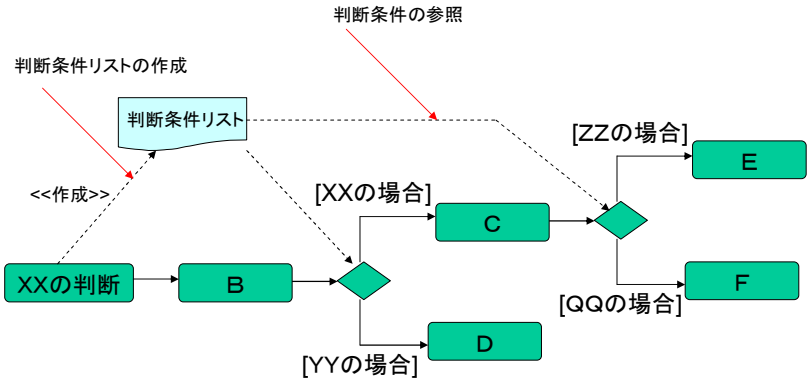


23

OPENTHOLOGY

ハイレベル手動による情報の入出力

- 判断条件を、情報として記録しておき、後で分岐条件として利用します。

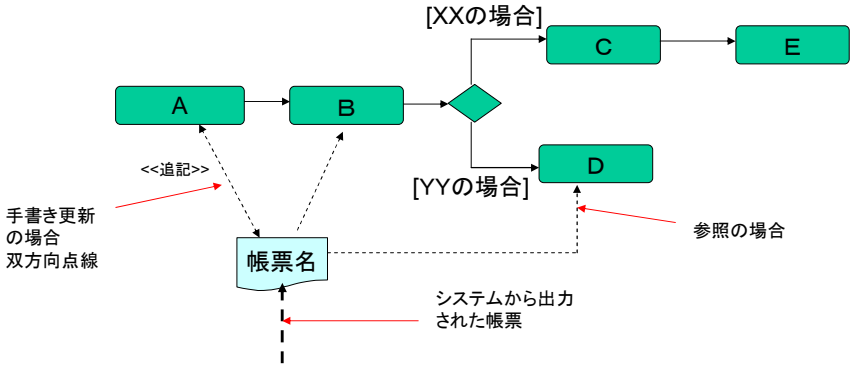


24

OPENTHOLOGY

ハイレベル　手動による情報の入出力

- システムから出力された情報に新たな情報を書き加える場合

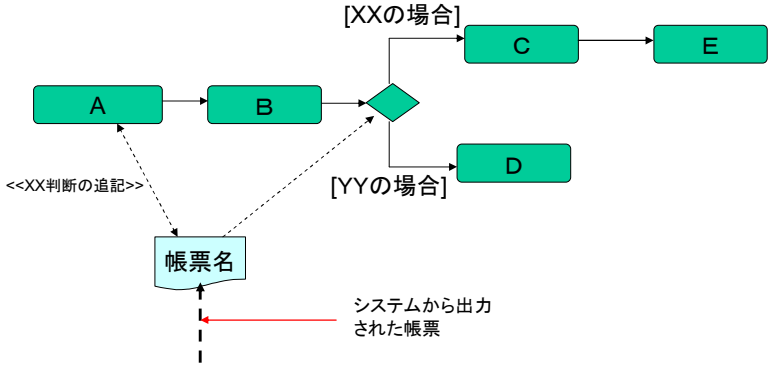


25

OPENTHOLOGY

ハイレベル　手動による情報の入出力

- システムから出力された情報に新たに判断条件を書き加えて利用する場合

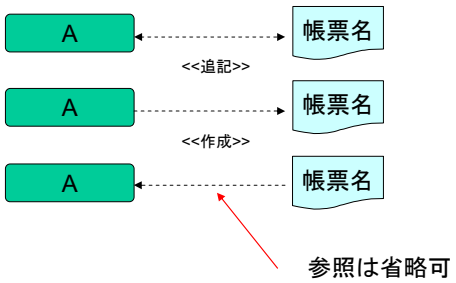


26

OPENTHOLOGY

帳票とアクティビティの関係
＜＜ステレオタイプの種別＞＞

- 帳票とアクティビティ



27

OPENTHOLOGY

業務モデリングの効果
＜現状分析の成果＞

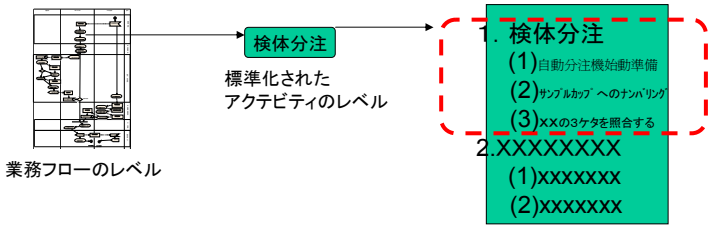
- 業務モデルを構成する業務要素（工程、アクティビティ、クラス）について、大まかな内容を得ることができます。
  - 従来は、細かな説明を読み理解してから、業務全体像を描く必要がありました。
- 標準化された工程とアクティビティを利用することで、誰が見ても理解できるフローを作成することができます。

28

OPENTHOLOGY

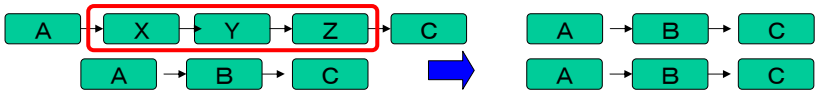
業務モデリングの効果
＜現状分析の成果＞

- 登場人物、全体業務フロー、アクティビティ、業務シナリオという順に業務を習得することができます。また、業務に変更が発生した場合は、必要とされるレベルで業務を把握することができます。
- いままでは、詳細な個々の業務処理の流れを理解するしかありませんでした。



粒度の調整

粒度を調整すると同じアクティビティとして認識できるものがでてきます



29

OPENTHOLOGY

業務モデリングの効果
＜戦略分析の成果＞

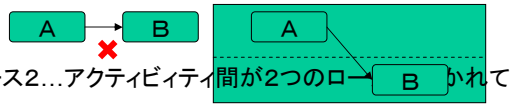
業務引継ぎや、業務転換コストの削減

工程とアクティビティ標準化されていくことで、社員共通の形あるナレッジとして業務パターンを蓄積できます。

業務過誤の防止

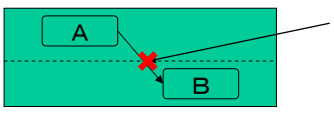
業務過誤が発生しそうな箇所を特定でき、議論できます。

ケース1…一人でやってしまうと間違いを起こしやすい連続した業務



ケース2…アクティビティ間が2つのロールに分かれている時

ロールを分割することを検討



・アクティビティの切り方が適切？
　ー責務の切り方として適切であるか？
　ー人で効率的に作業をやることができないか？
・引継ぎ資料に問題がないか？

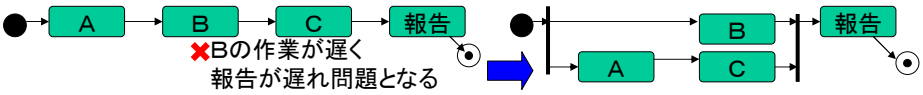
30

OPENTHOLOGY

業務モデリングの効果
＜戦略分析の成果＞

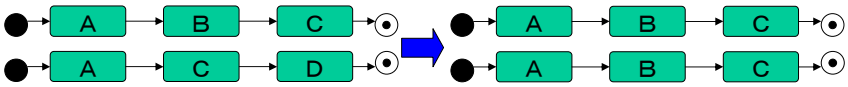
業務効率化・最適化を検討するベースとして

クリティカルセクションを見つけます。可能であれば並行性を検討します。

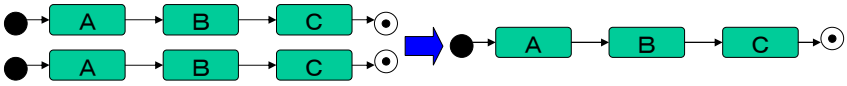


Bの並行処理を検討します。

業務のパターンを標準化します。



業務フローレベルで統一化します。（業務シナリオは異なるかもしれない）

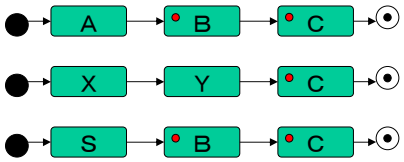


31

OPENTHOLOGY

業務モデリングの効果
＜戦略分析の成果＞

- 統一化された業務フローのレベルで、各業務に共通的なアクティビティを抽出マークをつけます。これにより、業務共通アクティビティの整理が可能となります。

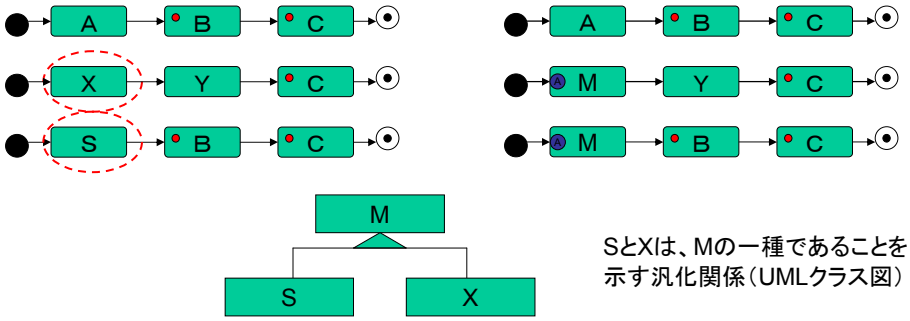


32

OPENTHOLOGY

業務モデリングの効果
＜戦略分析の成果＞

- 類似するアクティビティの共通点を取り出し抽象アクティビティを作成します。たとえば、XとSはMの一種だと分析すると抽象アクティビティMを作成し、抽象マークをつけ他と類別し、別途クラス図の汎化関係により説明します。



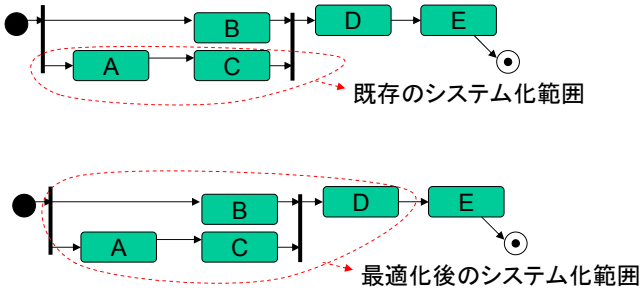
SとXは、Mの一種であることを示す汎化関係（UMLクラス図）

33

OPENTHOLOGY

業務モデリングの効果
＜IT化の成果＞

- 効率的な単位でIT化に結びつけます。

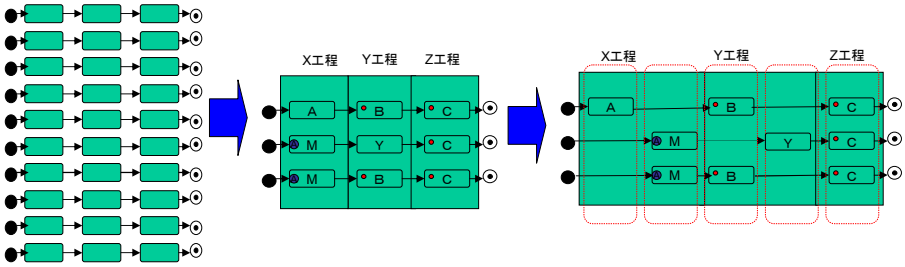


34

OPENTHOLOGY

業務モデリングの効果
＜IT化の成果＞

- システム開発は工程別、アクティビティ別に類別されます。共通的なアクティビティ共通部品として開発されます。



分析当初
13サブシステム

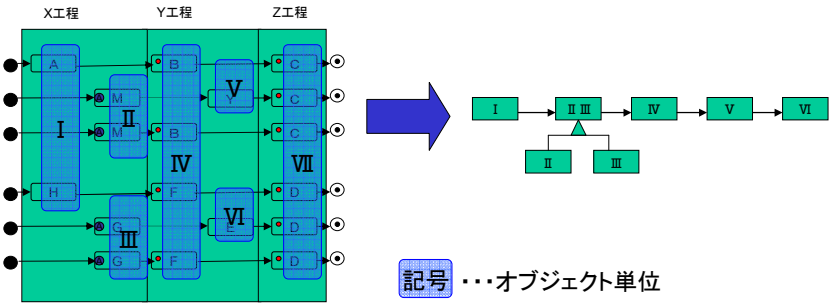
戦略分析後
6サブシステム

35

OPENTHOLOGY

業務モデリングの効果
＜IT化の成果＞

- オブジェクト（概念的なもの）という単位で作成された分析モデル（戦略フェーズで作成されるクラス図）を頼りに、アクティビティをオブジェクトの操作として整理します。
  - 業務構造をそのままソフトウェア構造に反映（分かりやすい）。
  - 保守の単位が明確になりやすい。
  - 複数のシステムの中でオブジェクトの再利用が進み、見通しのよい無駄のないシステムが構築可能



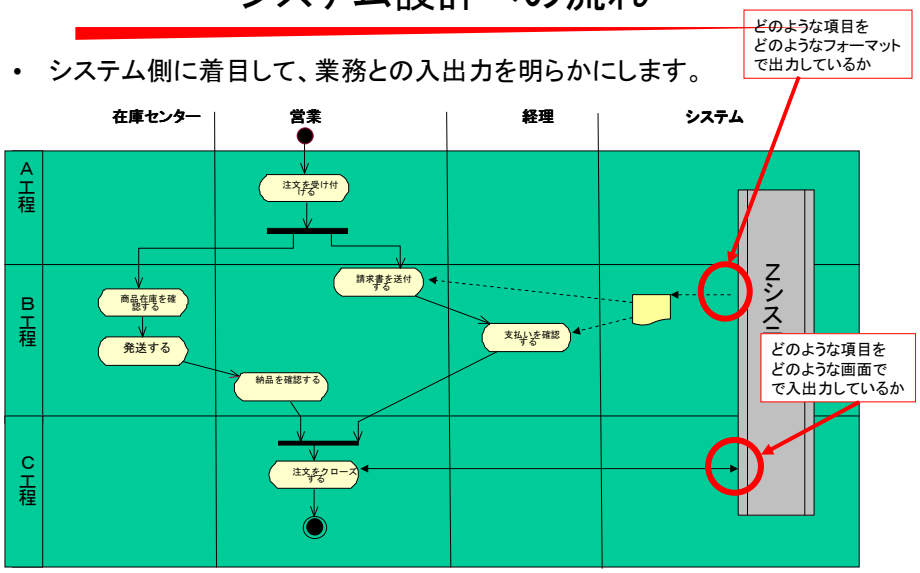
記号 ・・・オブジェクト単位

36

OPENTHOLOGY

システム設計への流れ

- システム側に着目して、業務との入出力を明らかにします。

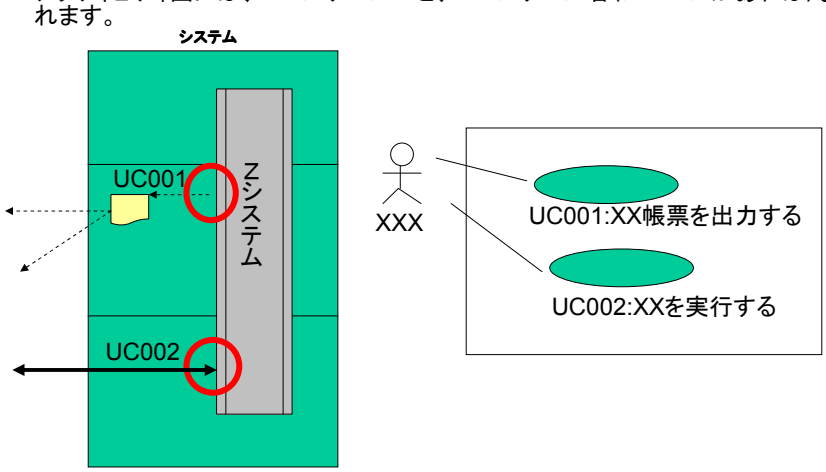


37

OPENTHOLOGY

ユースケース図の作成

- 参照、入出力ごとにユースケースとし、ユースケースモデルを作成します。
- アクティビティ図には、ユースケースIDと、ユースケース名（スペースがあれば）を入れます。



38

OPENTHOLOGY

ユースケース図の作成

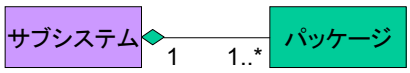
- ユースケース図の単位
  - サブシステム
- コンテキスト・ユースケース図
  - 3つのシステムをまとめた概念的なシステム境界

39

OPENTHOLOGY

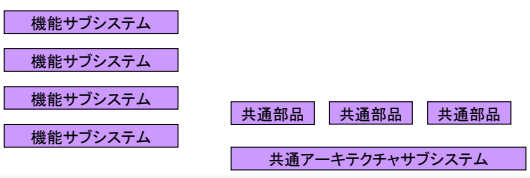
サブシステム

- サブシステムは、完結したサービスを提供する単位をいいます。
  - 旧来のものよりも粒度は小さめ
  - 必ず、**外部からの明確なサービスの窓口をインタフェース**として持ちます。
    - java の場合は、パブリックスコープのクラスをFacadeとして用意し、常にそこからアクセスする（他のクラスはパッケージスコープ）グループをサブシステム（コンポーネント）、パブリッククラスに誰でもアクセスできるグループを普通のパッケージ



サブシステム

1. ユーザから見たサービスの単位をグループ化
2. 共通アーキテクチャ、共通部品



40

OPENTHOLOGY

サブシステムビュー

- サブシステムは、論理パッケージ(UMLのパッケージ)にマッピングされます。

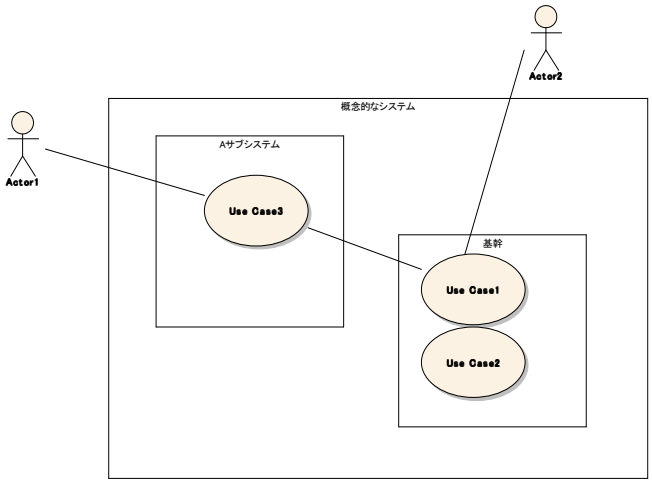
(Javaパッケージ)

41

OPENTHOLOGY

コンテキスト・ユースケース図

イメージ

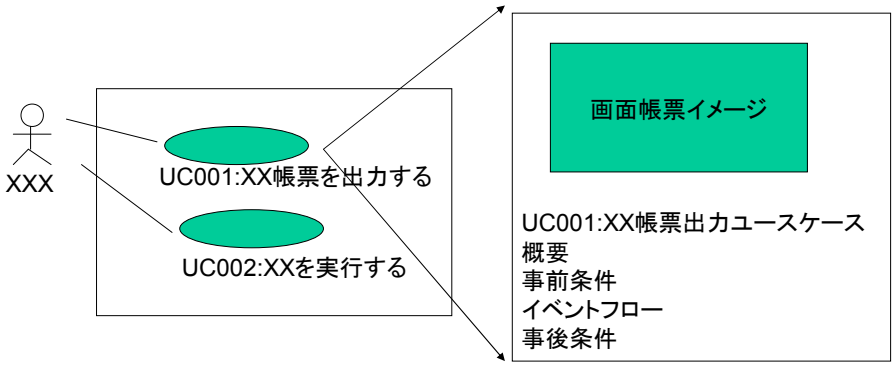


42

OPENTHOLOGY

ユースケース記述の作成

- 参照、入出力ごとにユースケースとし、ユースケースモデルを作成します。
- アクティビティ図には、ユースケースIDと、ユースケース名（スペースがあれば）を入れます。
- ユースケース記述が画面単位、帳票単位でOK



43

OPENTHOLOGY

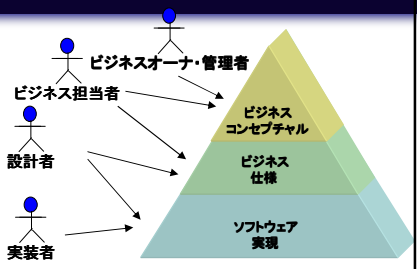
概念モデル編

Level1　コンセプチャルレベル

（参考）未完成
注）現段階では、例題が不適切です

OPENTHOLOGY

情報構造モデル（クラス図）



- **3段階に抽象度を分類**
  - **なぜ**
    - **ユーザナレッジを無理なく吸収できる**
    - **コンセプチャルレベルを持つことで、概念構造を分かりやすいレベルで視覚化・合意できる**

  - **Level1**
    - レベル…コンセプチャル
    - 用途　…現状分析モデル
    - 説明
      - 汎化、関連、集約などの関係を使って用語語彙の関係意味構造を表したもの。
  - **Level2**
    - レベル…仕様
    - 用途　…現状分析モデル、戦略分析モデル、システム分析モデル
    - 説明
      - クラスの責務を表現するために必要とされるキーや主な属性を割り付ける。また、責務を表す操作も微量ながら割り付ける。
  - **Level3**
    - レベル…実現
    - 用途　…システム設計モデル
    - 説明
      - データモデル
        - » データベースの論理設計につなげるため、主キー、副キーの明確化、すべての属性の洗い出しを行う。
      - 設計モデル
        - » 属性と操作について、実装アーキテクチャを意識した構造として割り付ける。また、実装アーキテクチャを意識した最適なクラス構造に変更する。

OPENTHOLOGY

業務概念の整理

- アクティビティ図
  - 業務処理をフロー（モデル）化したもの
  - アクティビティ図は、業務処理の変更に影響されやすい。業務処理が変わっても変化しにくいものをとらえると、より業務を理解しやすくなります。
- 業務概念
  - 業務に存在する普遍的な「概念」のこと
  - 業務概念の例
    - 検体、検査、顧客、検査項目

OPENTHOLOGY

業務概念＝オブジェクト

- 業務概念とはオブジェクトの事
- オブジェクトの見つけ方
  - 名詞や名詞句、動詞に着目する
    - 目に見える概念
      - 検体、病院、報告書、顧客
    - 目に見えない概念
      - セクション、グループ、測定原理

47

OPENTHOLOGY

業務概念＝オブジェクト

- オブジェクトの考え方
  - オブジェクトは、目に見えるもの、目に見えないものがあります。
  - オブジェクトは、同じ性質を持つオブジェクト（概念）がいくつか存在します。

オブジェクトはUMLでは
下線を引いて表します。

慶応病院
溝口病院
東京医大病院

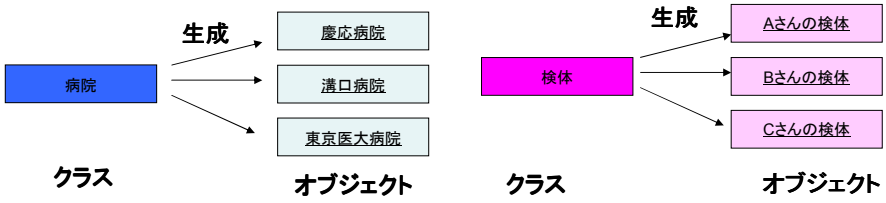
Aさんの検体
Bさんの検体
Cさんの検体

48

OPENTHOLOGY

業務概念＝オブジェクト

- クラス
  - オブジェクトの同じ性質を定義したもの
  - オブジェクトの設計図と考えます
  - オブジェクトはクラスの実体です

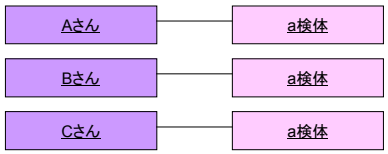


49

OPENTHOLOGY

関連

- オブジェクトには関係があります。
- 実は、「佐久間さんの検体」とは、佐久間さんという被験者と、検体との関係を表す概念です。



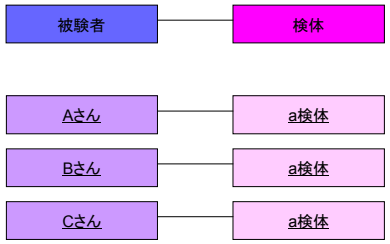
オブジェクト間の関係

50

OPENTHOLOGY

関連

- オブジェクトの関係は、クラス（オブジェクトの設計書）として表すことができます。
- これをUMLでは関連といいます。
- 関連は、オブジェクト間の関係の可能性を示します。
- オブジェクト間の関係をリンクと呼びます。



クラスで示す関係（関連）

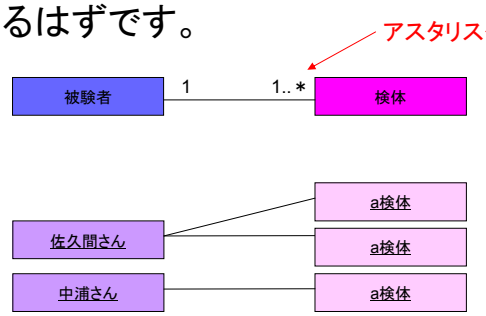
オブジェクト間の関係（リンク）

51

OPENTHOLOGY

関連の多重度

- 関連は多重度を持ちます。（クラスに書く）
- たとえば被験者は複数の検体を持つことがありますが、検体のない被験者は存在しません。一方、検体が存在するということは、かならず被験者とリンクしているはずです。



クラスで示す関係（関連）

オブジェクト間の関係（リンク）

52

OPENTHOLOGY

多重度の表記

- Aから見たB

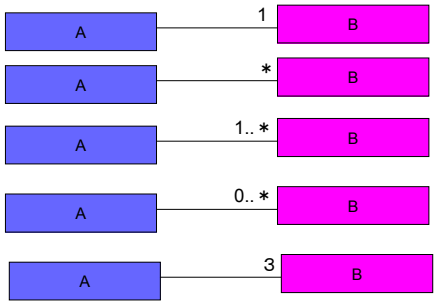
Aはかならず1個のBとリンクする

Aはいくつかの Bとリンクする

Aは1個以上のBとリンクする

Aは0個以上のBとリンクする

Aは3個のBとリンクする

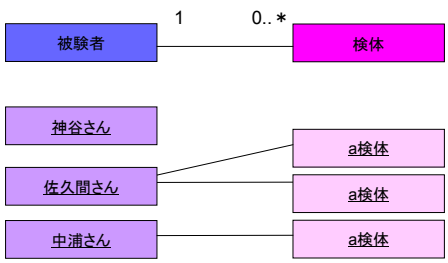


53

OPENTHOLOGY

関連の多重度(応用)

- 下記のモデルは、検体を持たない被験者の存在を許すことを表します。
- 関連の多重度
  - オブジェクト同士のリンクの可能性と制約を示すものです。



クラスで示す関係(関連)

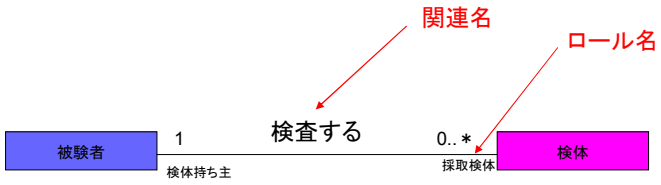
オブジェクト間の関係(リンク)

54

OPENTHOLOGY

関連の意味付け

- 関連名をつけることで関連の意味を深めることができます。
- ロール名をつけることで、参照先のオブジェクトから参照元のオブジェクトの役割を明確にすることができます。

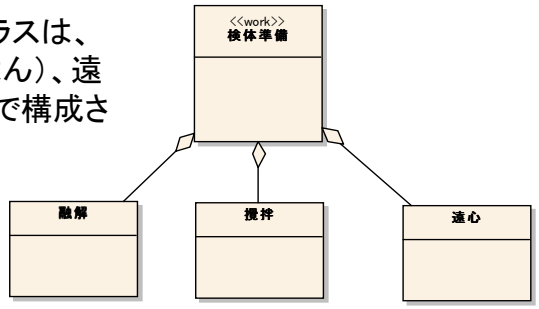


55

OPENTHOLOGY

集約

- 集約は関連の特殊系
  - 全体部分を表します。
  - 部分は全体の構成要素となります。
  - 検体準備というクラスは、融解、攪拌（かくはん）、遠心という準備要素で構成されています。

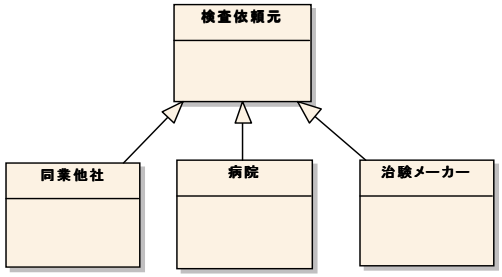


56

OPENTHOLOGY

汎化

- 関係の一種が汎化です
  - 下記は、検体依頼元とは、同業他社、病院、治験メーカを抽象化（汎用化）したものです。
  - 汎化は、似たようなクラス特性を見つけて定義することが重要です

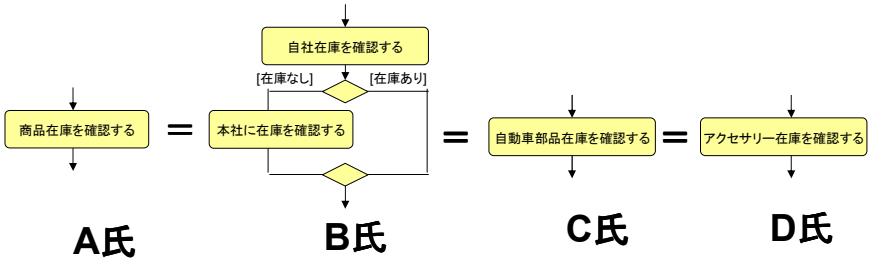


57

OPENTHOLOGY

集約と汎化をアクティビティ図で利用すると

- 業務チームによって作成されたアクティテビティがまちまちの時。どう整理するか？
- 単純にAの詳細化したものがB、C、Dか？



58

OPENTHOLOGY

アクティビティの関係

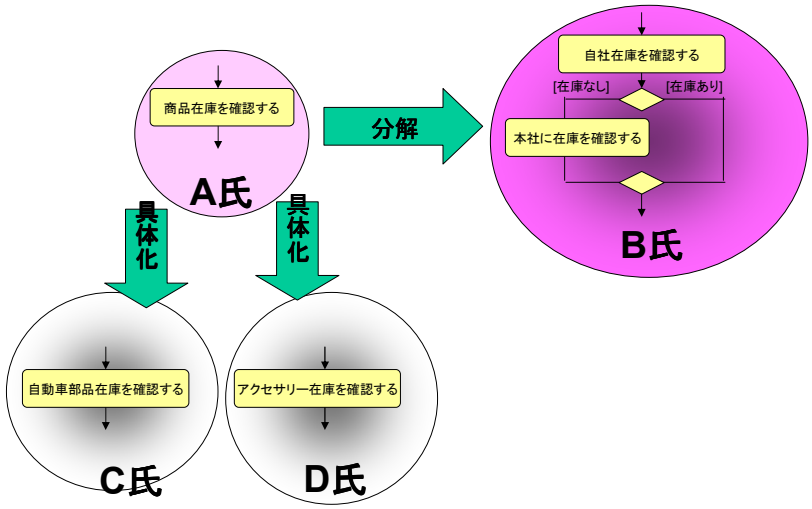


図6

OPENTHOLOGY

クラス図として表すこと

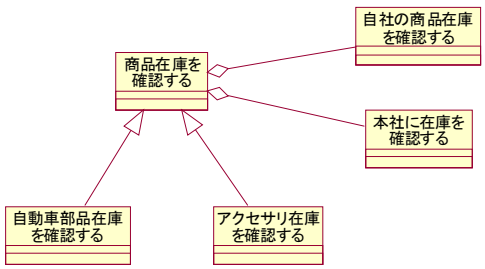
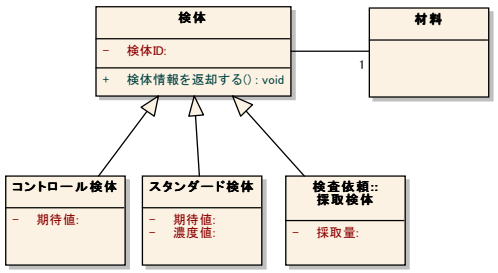


図7

OPENTHOLOGY

クラスの中身

- クラスには、複数の属性（データ）と複数の操作を定義することができます。中段に属性リスト、下段に操作リストを書きます。



61

OPENTHOLOGY

概念モデル編

Level2　仕様レベル

注）現段階では未完成です。

OPENTHOLOGY

概念モデル編

Level3　実装レベル

注）現段階では未完成です。

OPENTHOLOGY

モデリングガイドライン
＜＜ディスカバリーモデル＞＞

未完成ですが下記の
作成ガイドラインとなります

システム基本要求
（こうあるべき論）

問題解決リスト

問題分析ツリー

ビジョン反映リスト

ビジョン分析ツリー

OPENTHOLOGY

業務改善リストの作成

- 現状の問題
  - 現状のシステムの問題を簡潔に書く
- 改善方法
  - 改善方法について簡潔に書く
- 種別（業務 or システム or 両方）
  - 業務の改善かITシステムの改善か、両方関係しているか
- 関係部署
  - どの部署の改善であるか
- リスク
  - 改善に関わるリスクを記入
- 重要度（全体を見通した後記入）
  - どの程度重要かのランク付け
    - A(最重要)、B(重要)、C(普通)
- 改善コスト（全体を見通した後記入）
  - 改善を実施する上でのコスト（時間・工数）
    - A（すぐにやれる）、B（低コスト）、C（中コスト）、D（高コスト）

OPENTHOLOGY

改善内容

- 他部署との業務用語の標準化・統一化
- 視覚的な業務モデルを作成し利用する
- わかり辛い作業方法のやり方を改善する
- 非効率的な作業を見つけて効率化を図る
  - 自部門だけで解決できるもの
  - 他部門も含めて解決しなければならないもの
  - システム化すべきもの
    - 業務担当者とシステム担当者の問題に対する相互理解が必須
- 使いづらいシステムに対する改善要求
- システムの統一化によるコスト削減

OPENTHOLOGY

業務改善リスト

業務改善リストにより、最小コストで、最大効果を得るための道筋を見つける
　重要で、すぐに出来るものから
　　できるものは即実施

現状の問題点	改善方法	種別	関係部署	リスク	重要度	コスト	決定優先度	改善期間	関連資料

OPENTHOLOGY

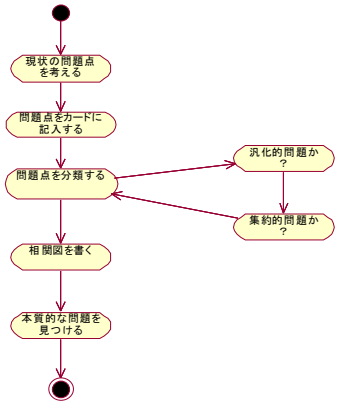
問題分析・業務改善の流れ

1. 既に明らかな問題点と改善方法により、業務改善リストを作成する
2. 問題分析ブレーン・ストーミングを実施
3. 問題改善ブレーン・ストーミングを実施
4. 2と3の結果により、業務改善リストの見直し、追加を行う
5. 業務改善スケジュールを作成する
6. 改善された後の業務フロー、概念モデルを再作成する

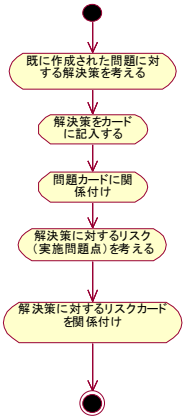
OPENTHOLOGY

問題分析と問題改善を考える

問題分析ブレーン・ストーミング



問題改善ブレーン・ストーミング



69