

宣教者(問題フレームの概要)

問題は「問題」にある

An Overview of Problem Frames

2009.6.18~19

はじめに
ジャクソン先生のこと
ジャクソン先生のアプローチ
重要な言葉の使い方
ドメインとインタフェース
イベントと共有現象
JSDのステップ
実世界のシミュレート
モデルプロセス

いくつかの教え
5つの問題フレーム
機械・ドメイン・要求
信号機の例
おわりに

J. Iizumi



はじめに

コンピュータの外側に本当の難しさがあるというのは、皆さんも感じられていることと思います。最近は、「様相」の重要性も噂されています♪

「問題フレーム」に注目している理由は、ソフトウェアの世界で本質的に難しい事柄に挑戦しようとしているからです。数学の世界でも、仮説をたてて証明すべき命題をうちたてるところが難しいですし、価値があります。ペレルマン(Grigory Yakovlevich Perelman)も偉いけど、ポアンカレ(Jules-Henri Poincaré)はもっと偉いですよね。少なくとも100年分の数学者の仕事を生み出しましたしね。

「問題フレーム」は著作(翻訳も)あるのですが、とても難解です。短い時間でお伝えするのは至難の技ですが、いくつかかいつまんで、ジャクソン先生(Michael Anthony Jackson)の「教え」をお話しようと思います

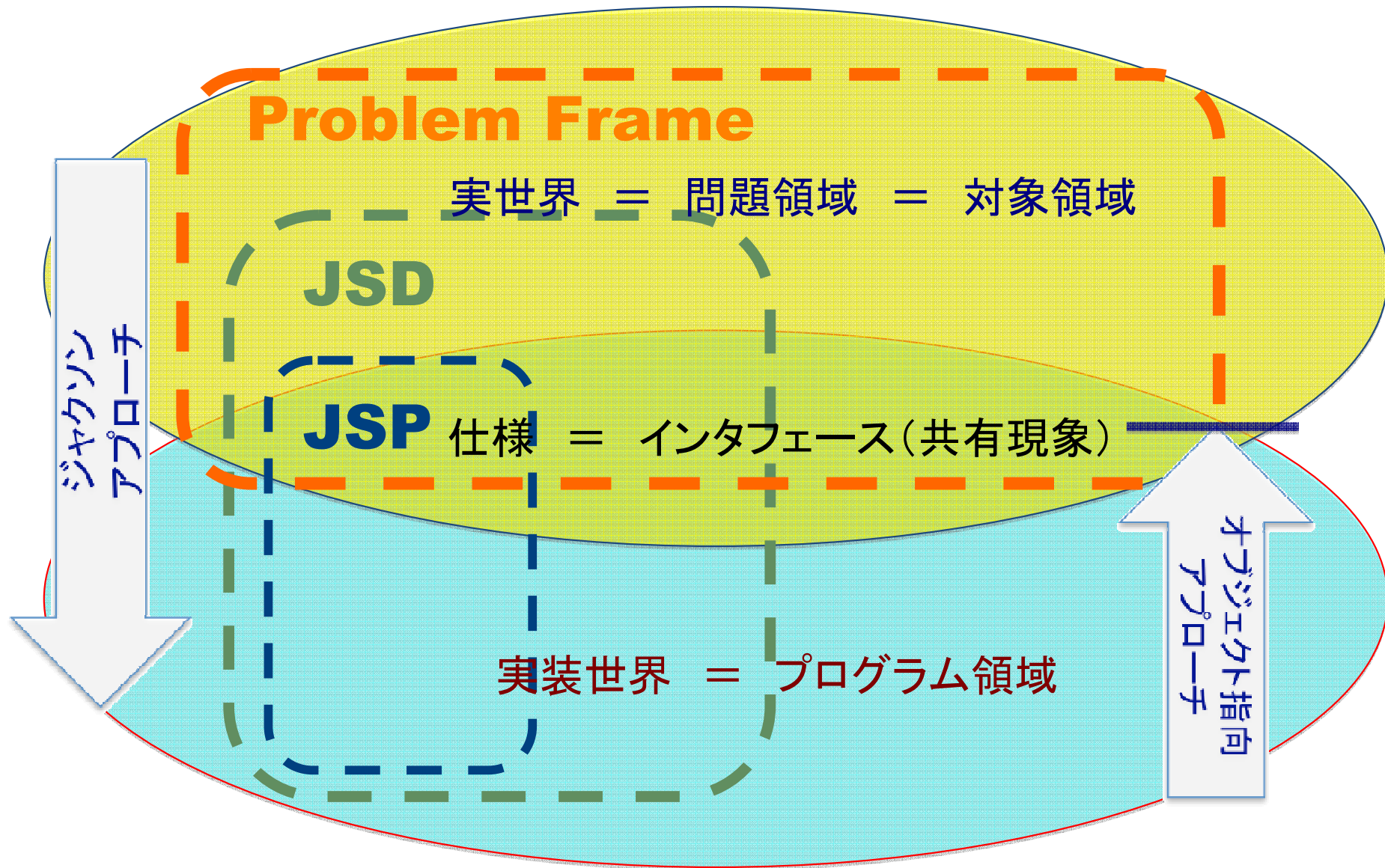
ジャクソン先生のこと

間違いなく、ソフトウェアエンジニアリング領域の奇才、というか巨匠。
哲学的な深さ、本質に迫る叡智、ただし、凡人にはかなり判りにくい。

3つの奇跡

- ❖ 奇跡1 : JSP(Jackson Structured Programming), 1975年出版
 - プログラム開発手法(構造化プログラミング+構造不一致)
- ❖ 奇跡2 : JSD(Jackson System Development), 1983年出版
 - システム開発手法(ユーザは要求を知らない。機能より実世界モデル)
- ❖ 奇跡3 : 問題フレーム(Problem Frames), 1995年, 2001年出版
 - (システムから遠い)実世界の問題の分析方法

ジャクソン先生のアプローチ



重要な言葉の使い方

❖ システム

➤ 広義のシステム

- 世界とコンピュータとを一緒にした全体
- 人間や業務だけのシステムも考えることができる

➤ 狭義のシステム ⇒ 「機械(マシン)」

- コンピュータとそのソフトウェア

❖ モデル

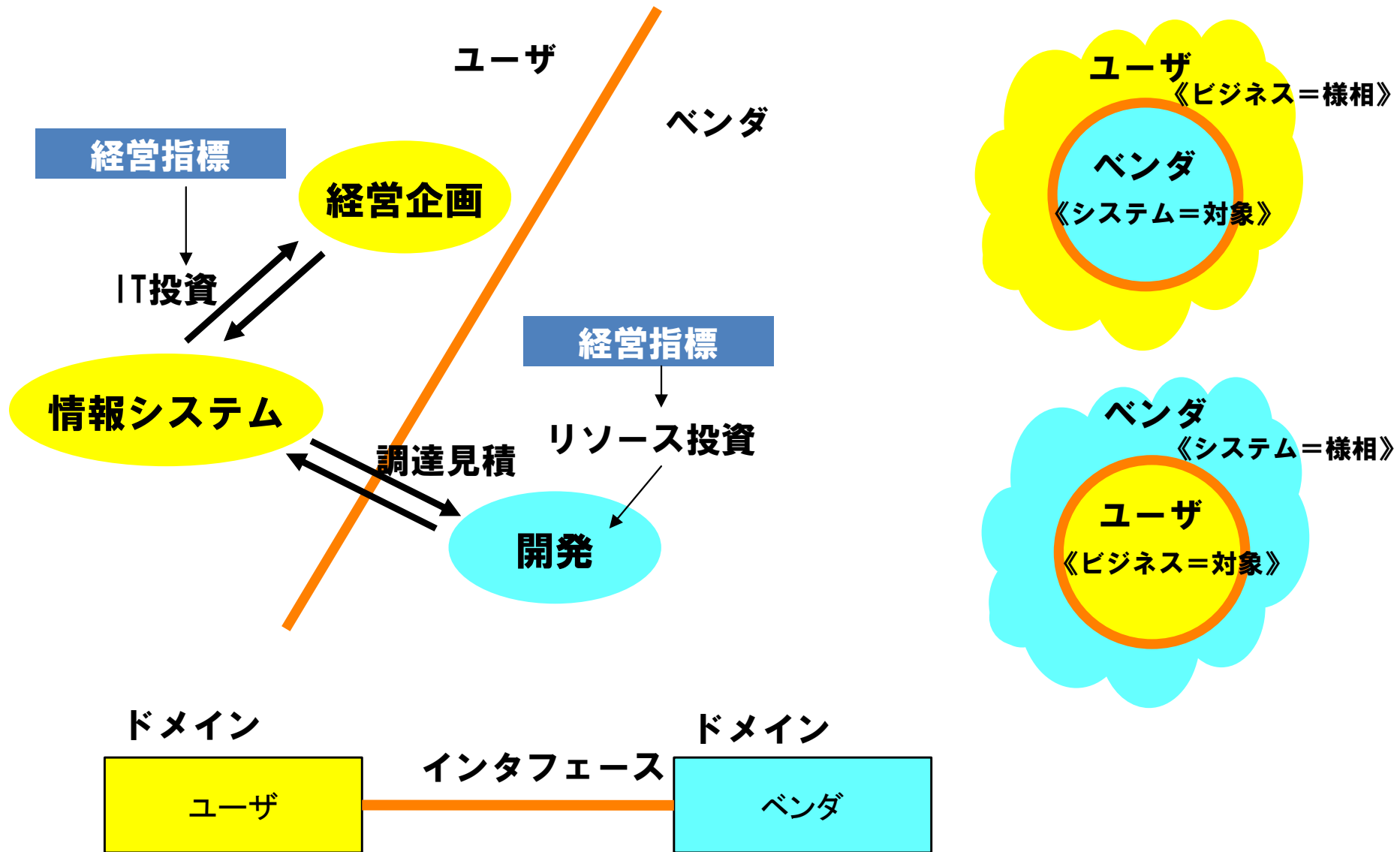
➤ 分析モデル ⇒ 「記述」

- 世界の記述
- 経済に関する微分方程式、振舞いの記述等
- UMLモデルは、UMLという規約言語による「記述」

➤ 類推モデル ⇒ 「モデル」

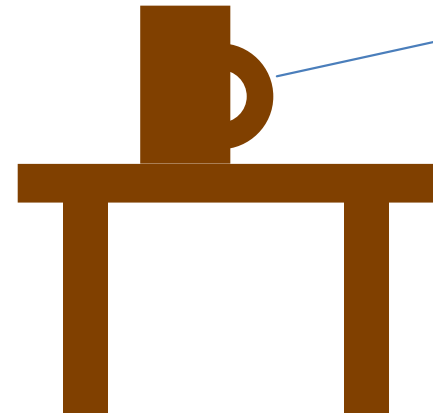
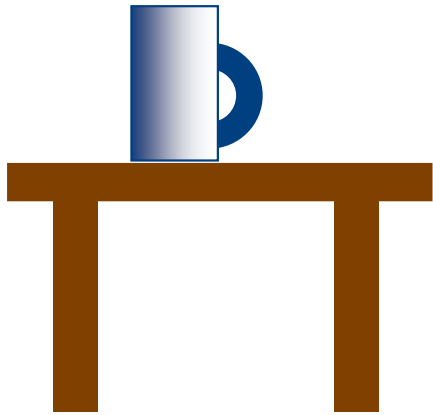
- 類似の特性を持った別の実体

ドメインとインタフェース(共有現象)



イベントと共有現象

ものごとを認識する基本は《イベント》＝変化＝差異＝一瞬です。



これは机の
一部かも

共有現象（インタフェース）としてのイベントを「共有事象（common action）」といい、これを抽出することがシステム設計の第1歩です。

人の生涯



人が時計のボタンを押す

共有事象

時計が人にボタンを押される

時計の生涯

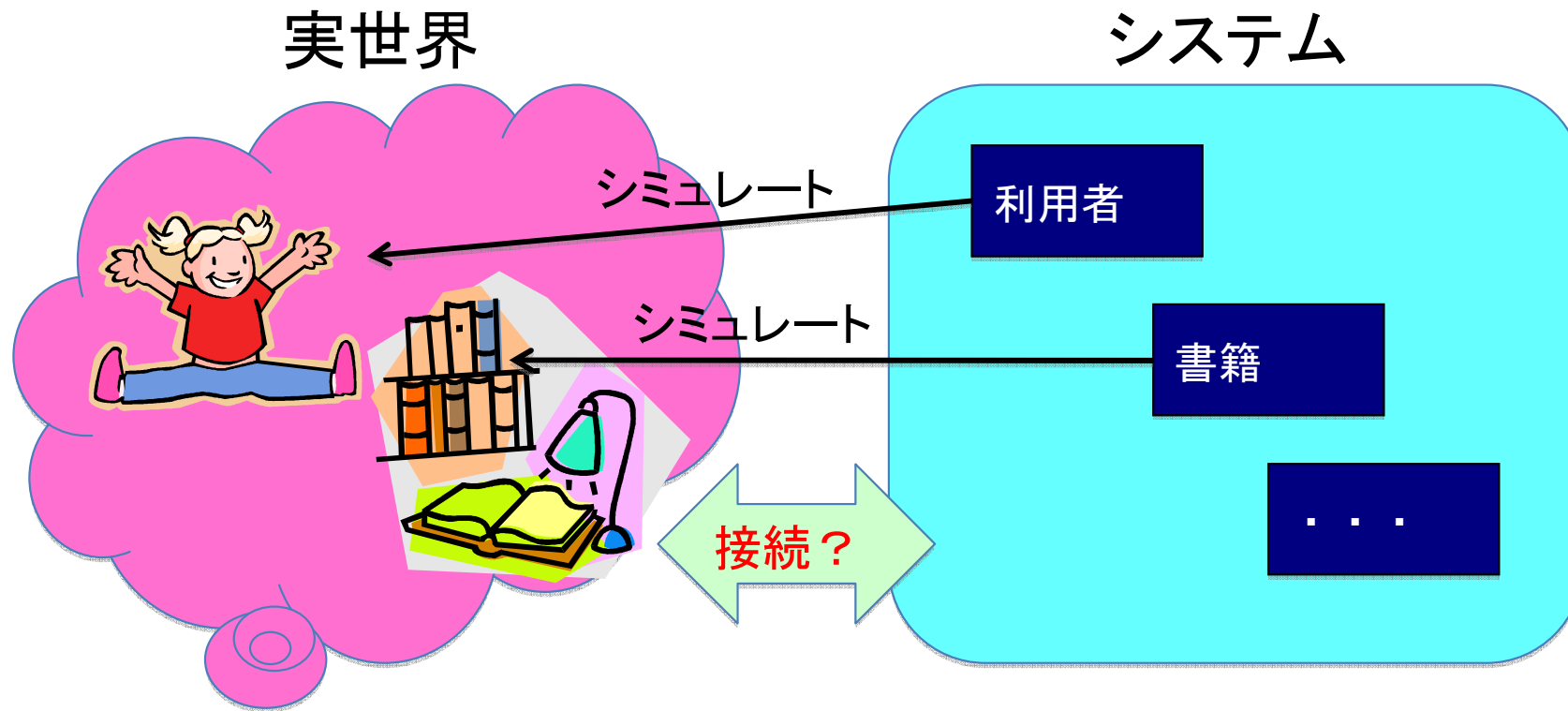


JSDのステップ

- ❖ JSDステップ1: 実体・動作抽出 (Entity Action Step)
 - 実世界の实体とその動作を抽出して記述する。
- ❖ JSDステップ2: 実体構造化 (Entity Structure Step)
 - 実世界の動作の順序関係を3基本構造で構造化する。
- ❖ JSDステップ3: 初期モデル作成 (Initial Model Step)
 - 実世界の实体に対応したシステム内のモデルプロセスを作成する。
- ❖ JSDステップ4: 機能定義 (Function Step)
 - 実世界の動作が起こった時のシステムの機能プロセスを定義する。
- ❖ JSDステップ5: タイミング制約 (Timing Step)
 - 時間的な制約を(インフォーマルに)記述する。
- ❖ JSDステップ6: 実現 (Implementation Step)
 - 物理的なプロセッサ数で実行可能になるようにモデルと機能プロセス構造を変換する。

仕様化(specification): 要求定義に相当

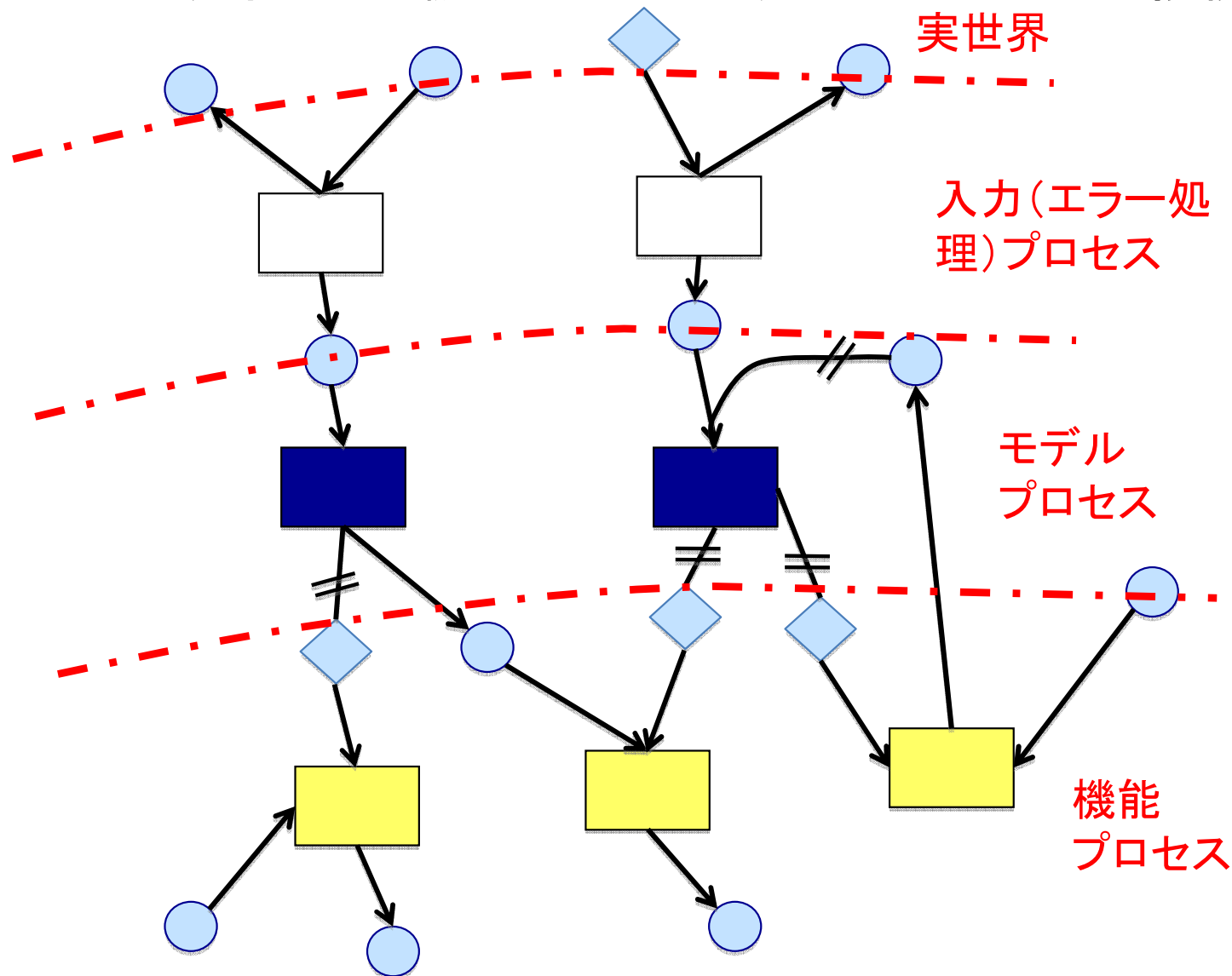
実世界をシミュレートする



- ❖ 実世界のある「書籍」が貸出されたら、システム内にあるモデルプロセスBookの該当するものが、同じ動作をする
- ❖ そのためには、実世界とシステムとをどのように「接続するか」を決めなくてはならない

モデルプロセス (JSD:ステップ4の結果)

- ❖ 一般的に、ステップ4の結果は、以下のような形になる
 - ステップ4で定義される機能プロセスは、モデルプロセスに接続される



いくつかの「教え」(額縁用)

実世界にある問題を、ソフトウェアを使って解く

解くべき問題の範囲のことを「コンテキスト(文脈)」と呼ぶ

コンテキストは、
どこに(where)という観点で物理的ドメインを描く

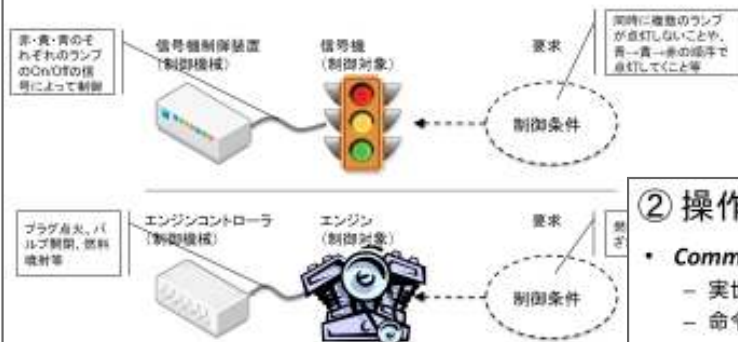
要求とは、機械が及ぼす効果によって
適用領域において達成されるべき条件

5つの問題フレーム（問題のパターン）で分析できちゃう♪

① 動作制御フレーム

• Required Behavior Problem Frame

- 実世界に、動作を制御すべき対象があり、
- これを制御する機械を構築する



② 操作命令フレーム

• Commanded Behavior Problem Frame

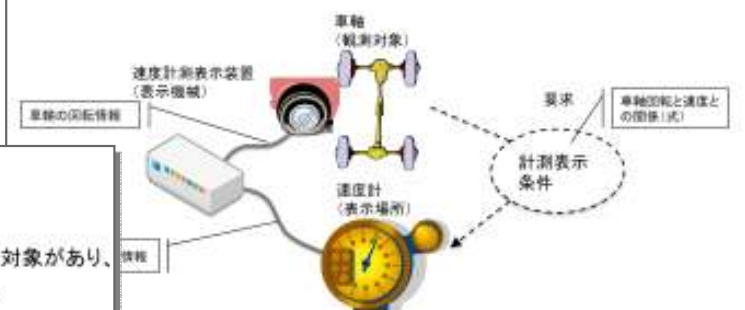
- 実世界に、操作者が発行する命令に従って動作を制御すべき対象があり、
- 命令を受け、それによって対象を制御する機械を構築する



③ 情報表示フレーム

• Information Display Problem Frame

- 実世界に、継続的に状態や振舞いを観測すべき対象があり、
- その情報を収集し、必要な形式で必要な場所に提供する機械を構築する



④ 単純編集フレーム

• Simple Workpieces Problem Frame

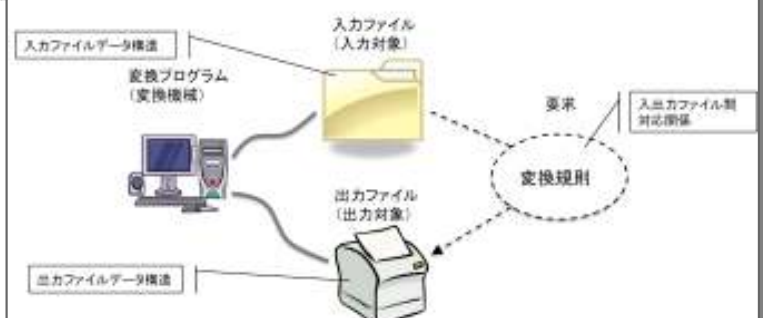
- コンピュータで処理可能な、文書、図形、画像等を対象とし、
- これを編集する機械を構築する



⑤ 変換フレーム

• Transformation Problem Frame

- コンピュータで読み込み可能な入力ファイルを、
- 要求される規則に従って出力ファイルに変換する機械を構築する

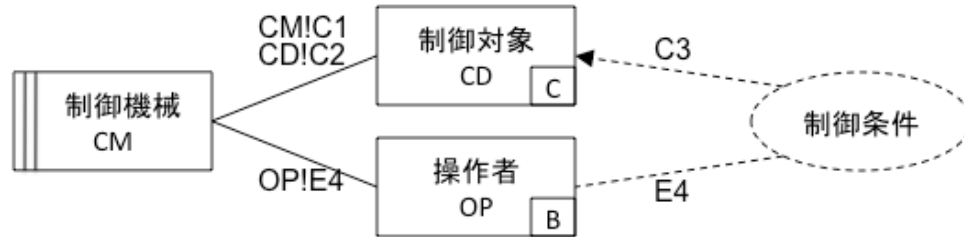


機械(マシン)とドメイン(適用領域)と要求

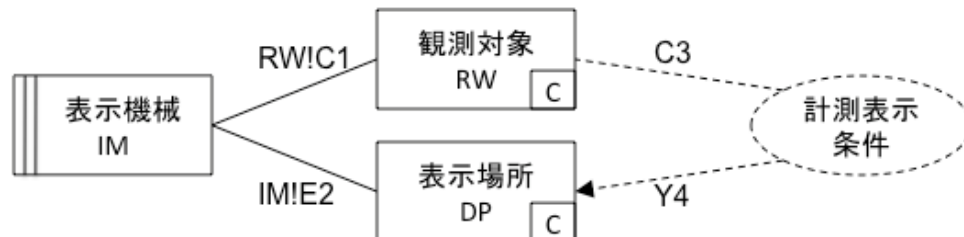
① 動作制御フレーム: Required Behavior Problem Frame



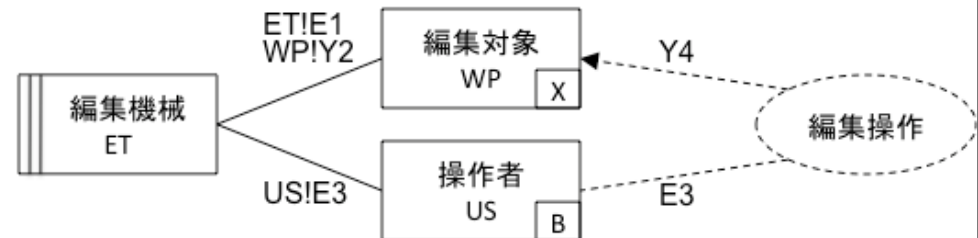
② 操作命令フレーム: Commanded Behavior Problem Frame



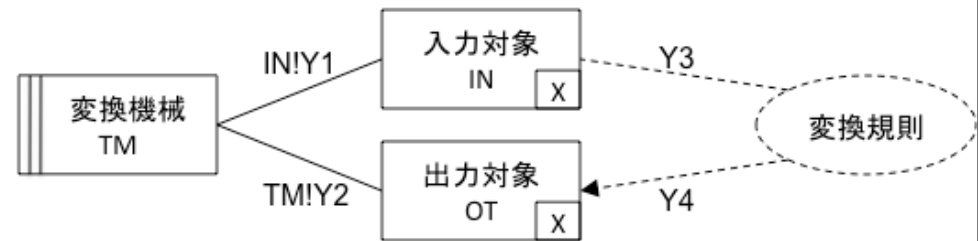
③ 情報表示フレーム: Information Display Problem Frame



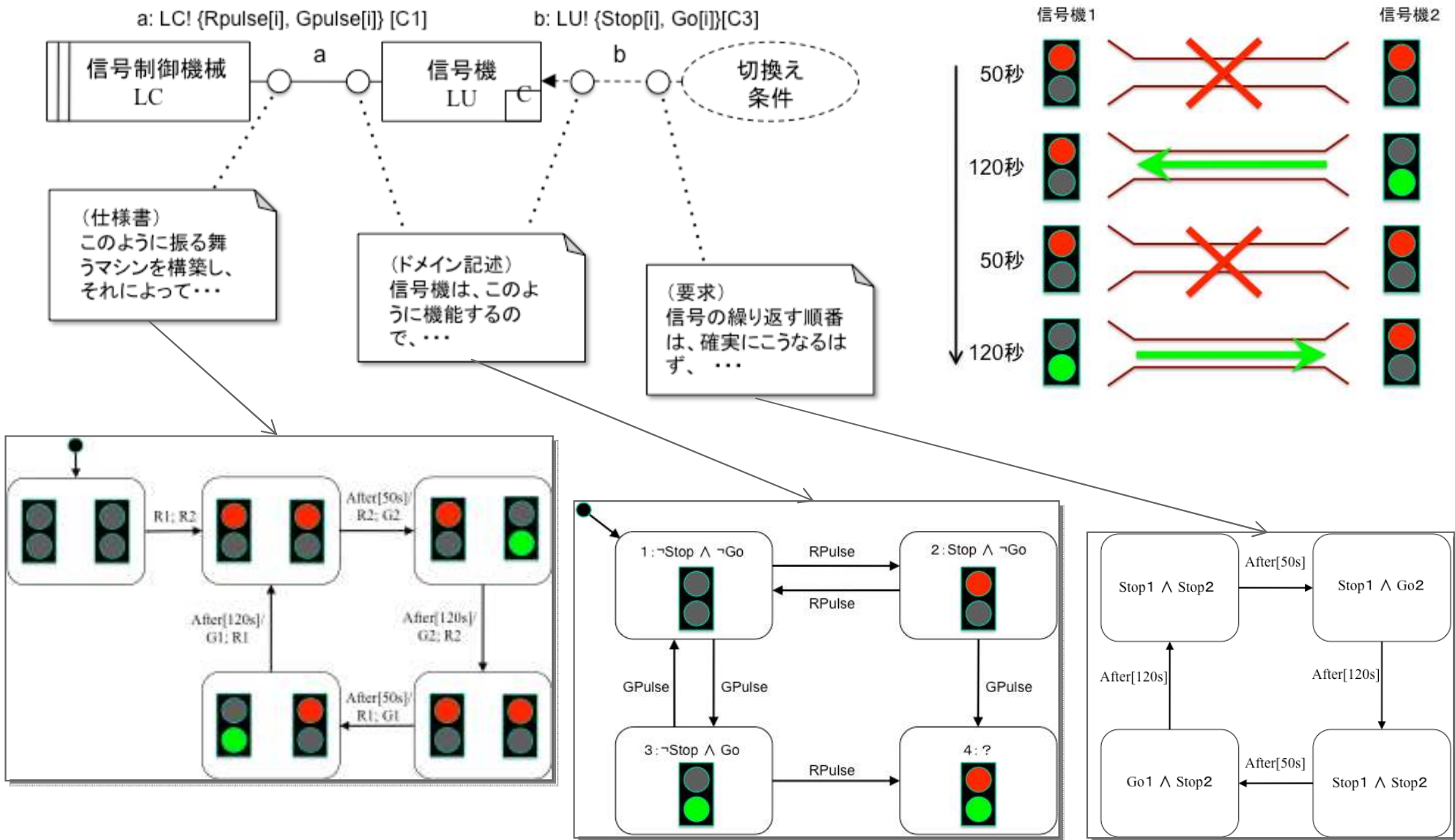
④ 単純編集フレーム: Simple Workpieces Problem Frame



⑤ 変換フレーム: Transformation Problem Frame

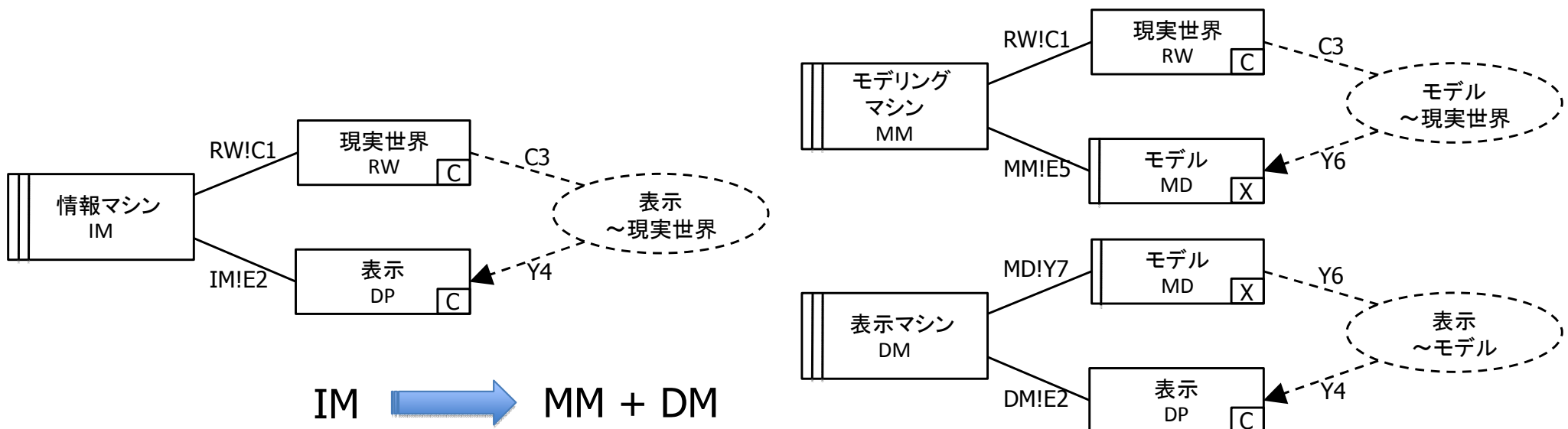


信号機の例（動作制御フレーム）



モデルドメインと現実世界

- ❖ モデル: 類似モデル
情報問題の現実世界ドメインのこと
- ❖ モデルドメインは、情報機械内部の非共有現象の一部、すなわち、表示出力の計算に使用する変数のセットを分離して明確にします。
- ❖ モデルドメインを使用する際には、問題は2つの副問題: モデル構築部分と、モデル利用部分とに分けられます。
- ❖ モデルドメインの役割は、過去を要約すること、現実世界の推論を裏付けること、非常に長い計算の結果を予想すること等があります。
- ❖ モデルドメインは、現実世界とは乖離する可能性もあります。



おわりに

なお、余談ですが、無謀にも「問題フレーム」を民間セミナーでもやっていますので、是非周辺の方々にもご案内いただければ幸甚です。

<http://www.catnet.ne.jp/triceps/src/0730%2029-250.htm>

