モデル検査ブートキャンプ B4-1

支援ソフトウェアを活用した実践的モデル検査

2008年1月30日

関西電力株式会社
メルコ･パワー･システムズ株式会社

発表の流れ

モデル検査による問題解決に必要な実践力の理解

１．はじめに

２．モデル検査の対象選定

３．モデル化の留意点

４．モデル検査器の実施

５．反例解析

６．まとめ

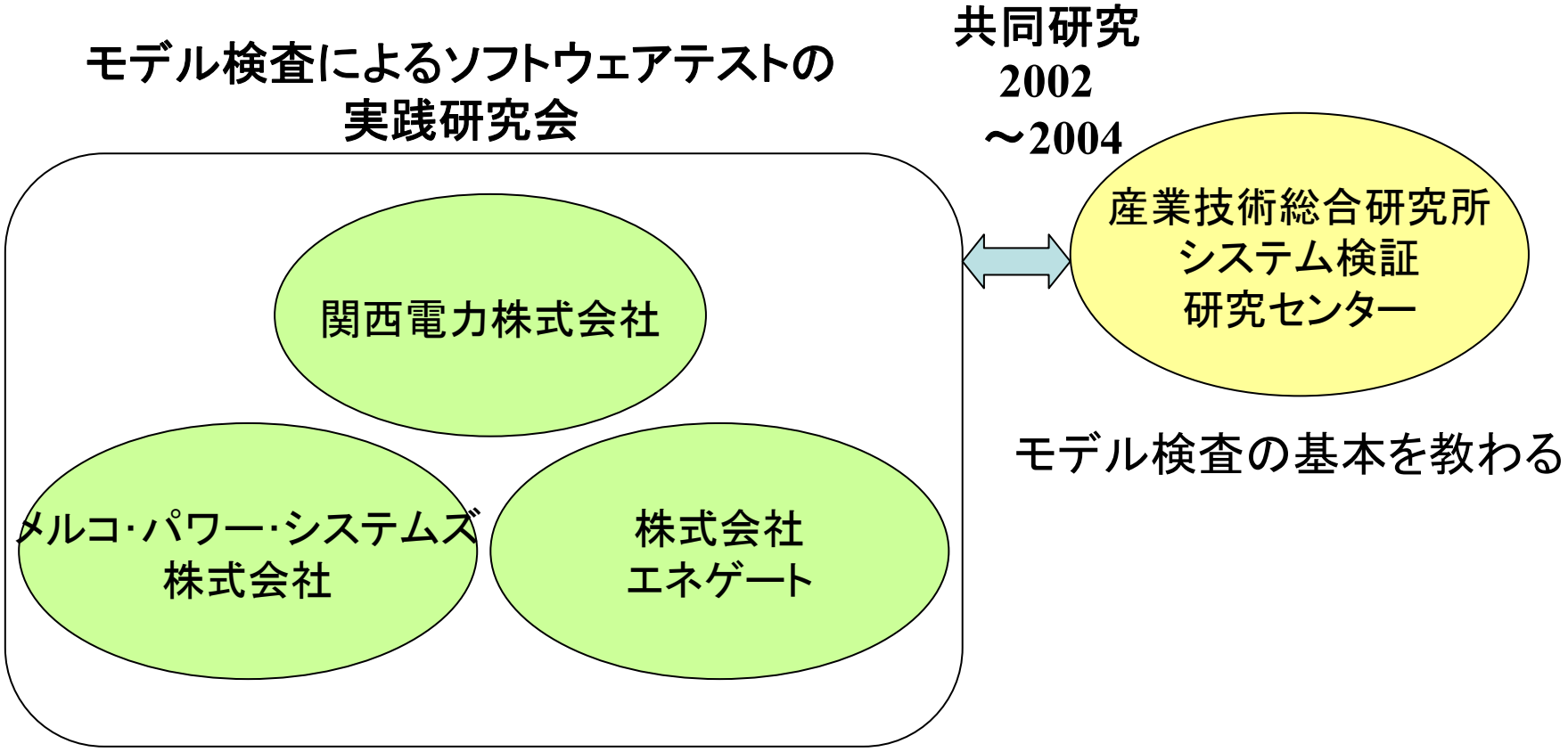
現状認識

戦略

戦術

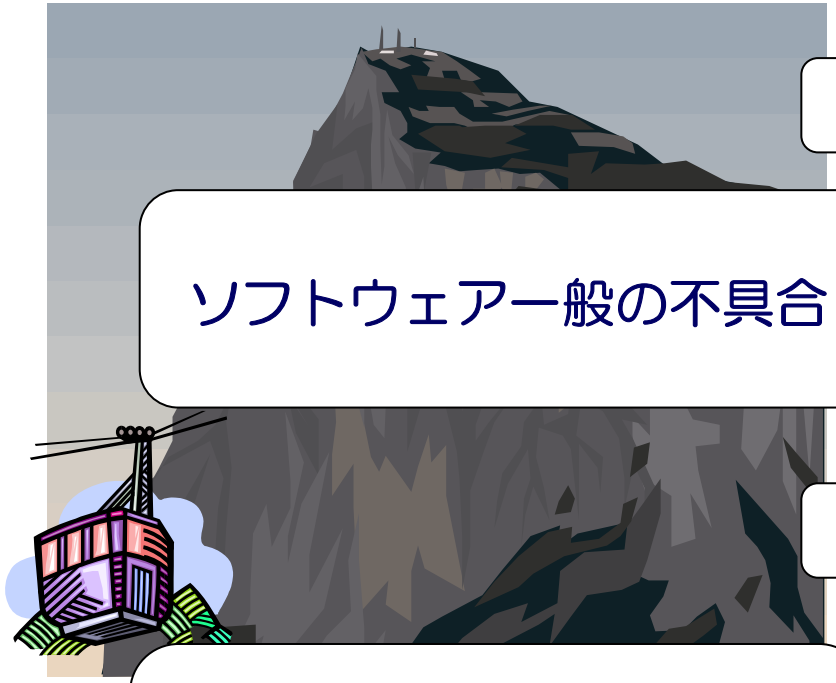
実際の流れ

背　景



３社合計で２０件以上のシステムにモデル検査適用

実用化の方向性

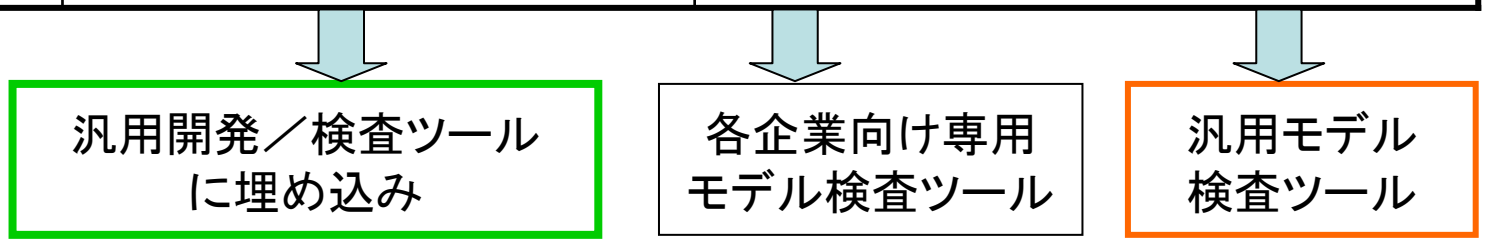


	ソフトウェア一般の不具合	システム固有の問題に入り込んだ不具合
ねらい		
特　徴	・誰でも使える ・自動的にモデル化 ・モデルを意識しない	・システムの知識が必要 ・人手によりモデル化 ・モデルをしっかり考える

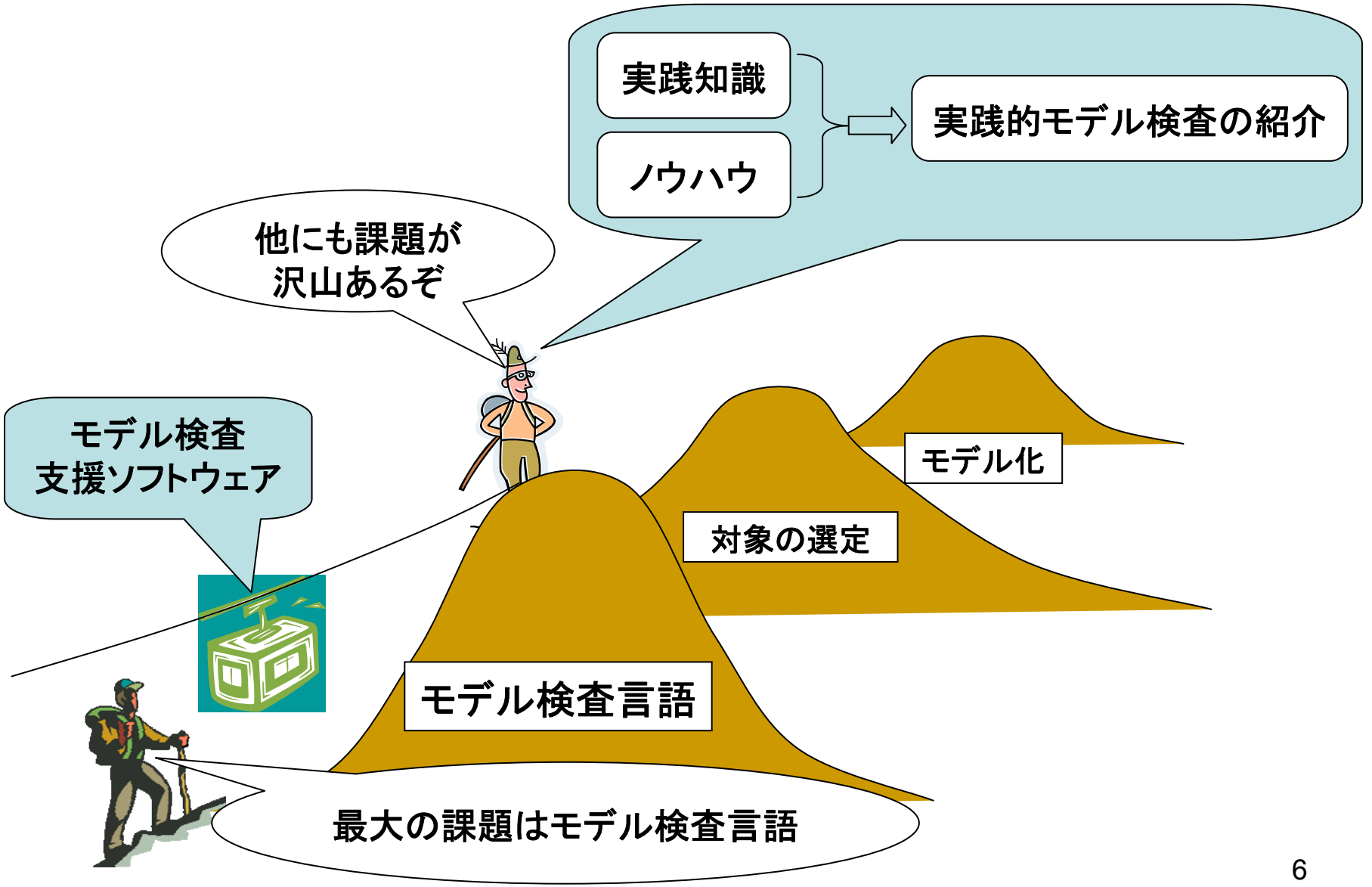
・分野、対象を限れば、自動化は可能

実用化の方向性

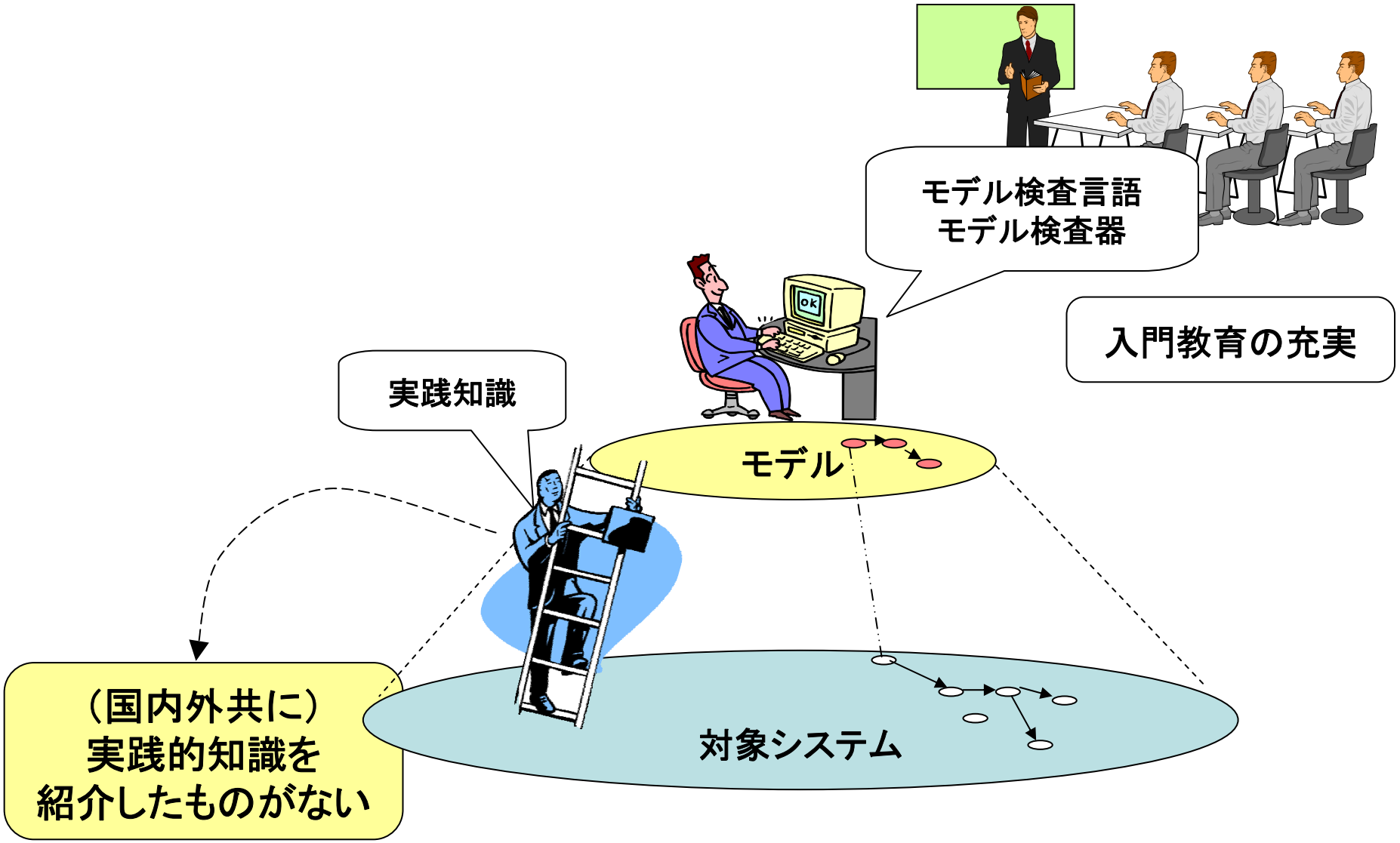
ねらい	ソフトウェア一般としての品質向上	対象システムに固有な品質の向上
検査項目	汎用的	個別
検査式の作成	不要（予め準備できる）	必要（対象に合わせて作成）
モデルの汎用性	あり	なし
モデルの作成	自動的なモデル化／抽象化が可能	対象に合わせたモデル化／抽象化が必要
検査の自動化	比較的容易	難しい
学習の必要性	不要	必要
学習のレベル	なし	モデル検査の考え方～モデル化の技法、 作業のノウハウ



実用化へのハードル


実践知識
ノウハウ
実践的モデル検査の紹介
他にも課題が
沢山あるぞ
モデル検査
支援ソフトウェア
モデル化
対象の選定
モデル検査言語
最大の課題はモデル検査言語

技術者向け教育の現状


モデル検査言語
モデル検査器
入門教育の充実
実践知識
モデル
対象システム
（国内外共に）
実践的知識を
紹介したものがない

モデル検査の対象選定

実践では、できるだけ短時間で期待する成果が求められる

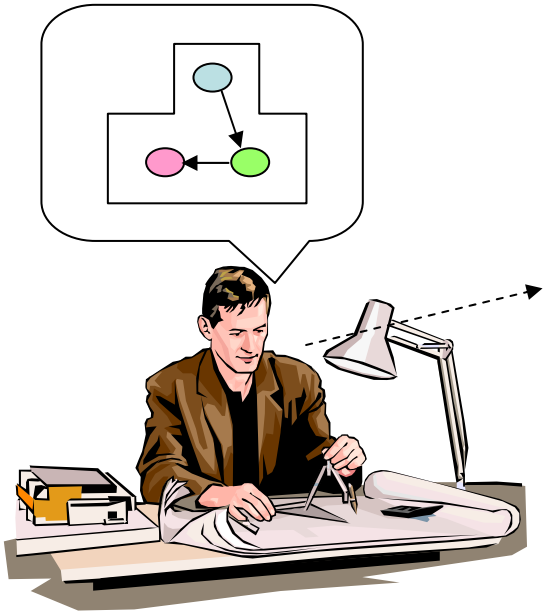
開発システムのどの部分の
何をモデル検査するのか？

- 対象システム／検査内容は
  モデル検査に適しているか？
- 検査すべき項目は何か？
- モデル検査の結果は、価値が
  共有できるか？

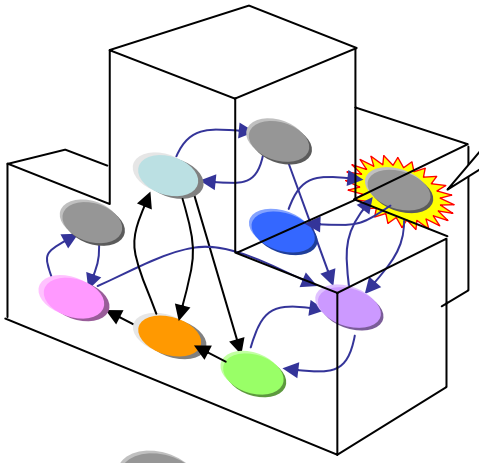
モデル検査の向き不向き

適性： 状態遷移モデルの全パス検査が有効なもの

設計者の視点からの認識



ソフトウェア設計
【起こり得る動き】



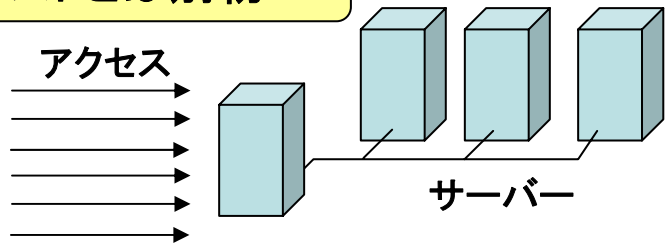
：見落としている状態

人間がトレースしきれず
見落としてる危険状態

発見

・組込みソフト
・監視制御ソフト
・Webアプリの画面遷移
他で実績

負荷テストとは別物

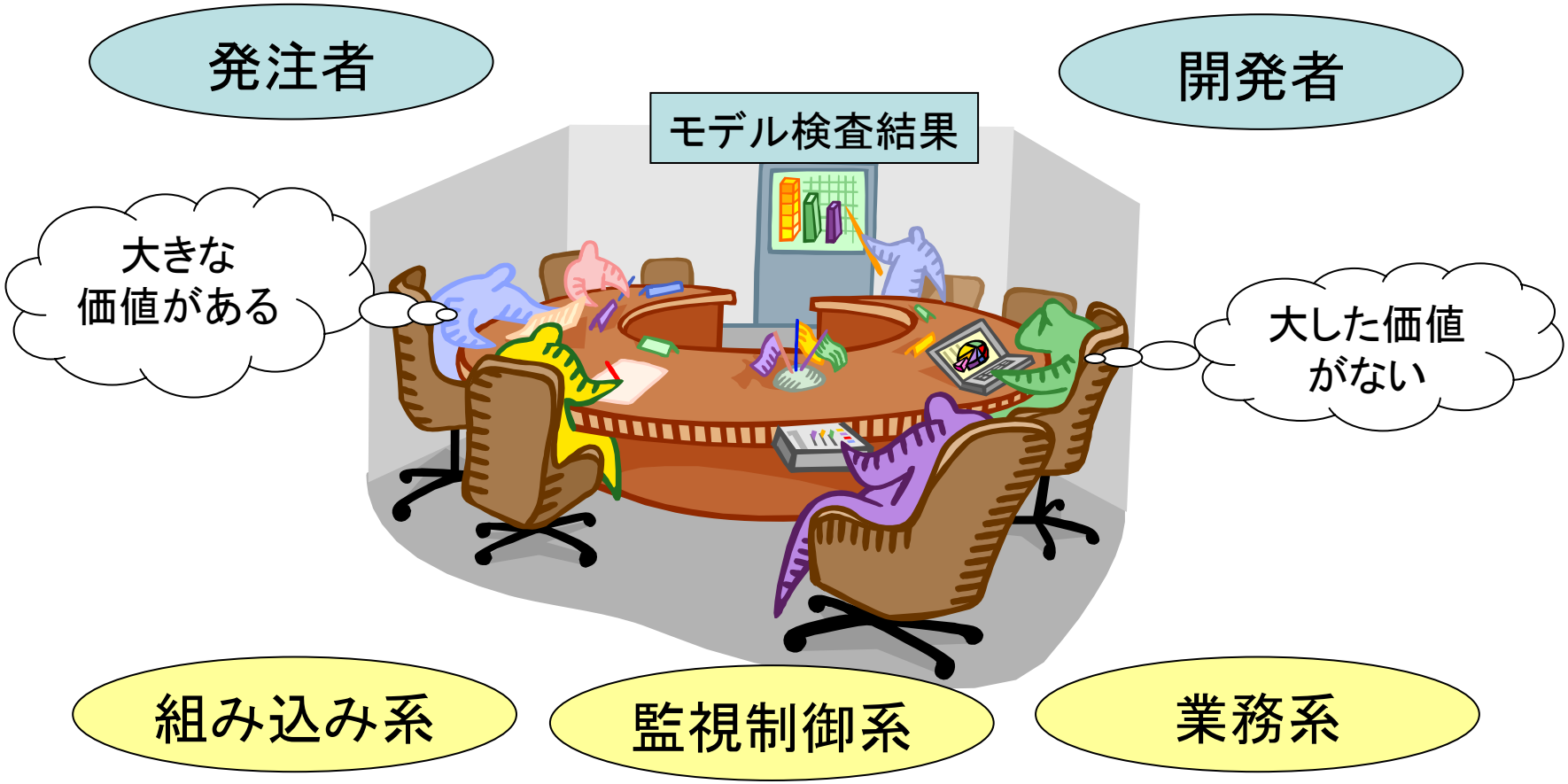


工程別の特徴

対象	モデル規模	所用時間	特長
仕様書	中	中	検査の目的、外部環境に注意してモデル化要
設計書	大	長	規模によっては、モデルの分割、組合せを行う
コード（デバッグ）	小	短	検査目的に応じて大幅にモデルを縮小

評価が得られる（価値が共有されている）

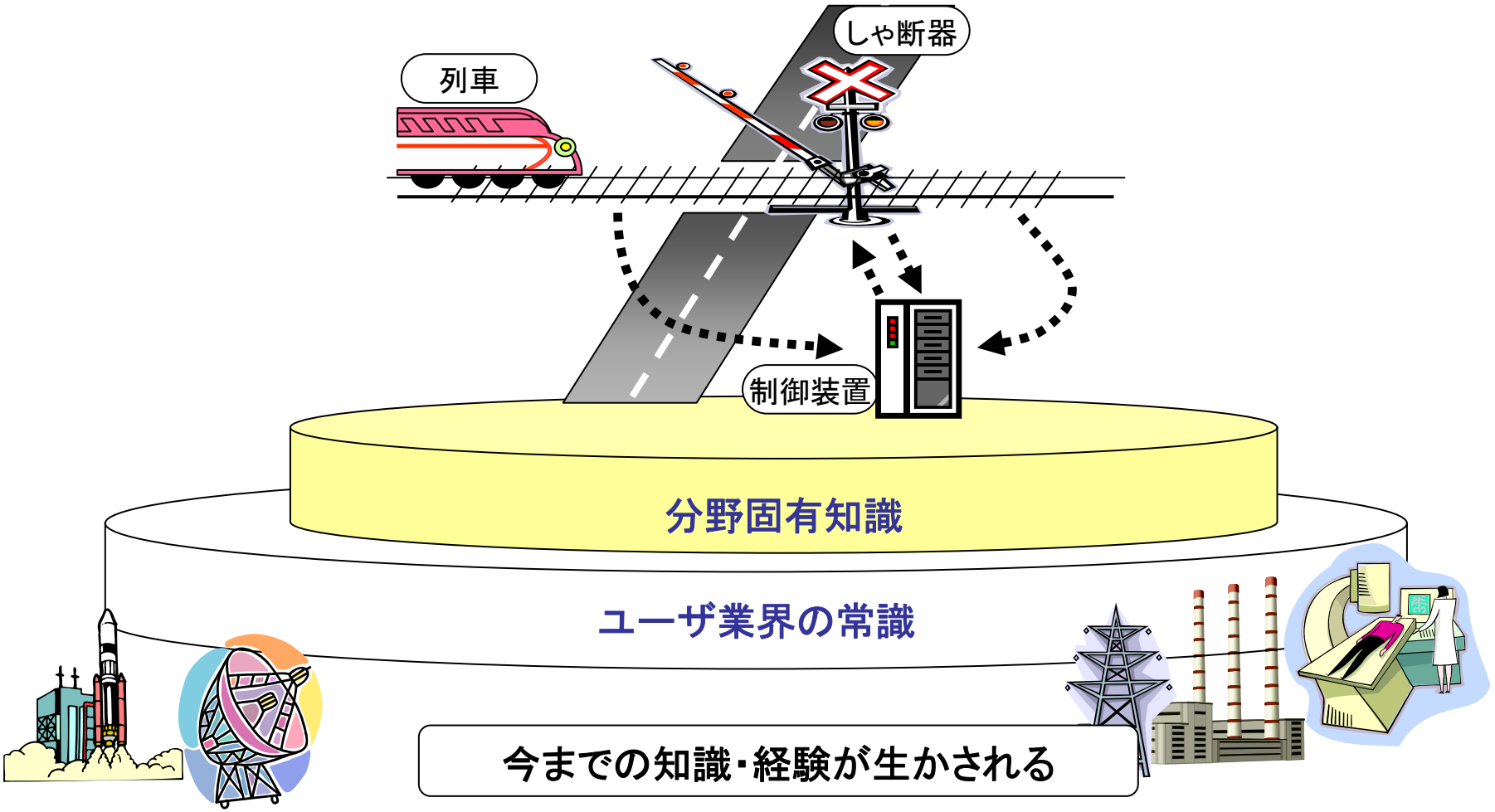
何が大事か立場によって違う⇒価値の共有できるもの



検査すべきこと⇒固有知識が不可欠

起きてはいけないことは何？

出来ていなければならないことは、何？



固有知識＋アルファ

個々の動作よりも全体として
満足／保証すべきことは何？

従来とは視点を変えてみる

見落としに気付かない

個別の要求は、間違いにくい

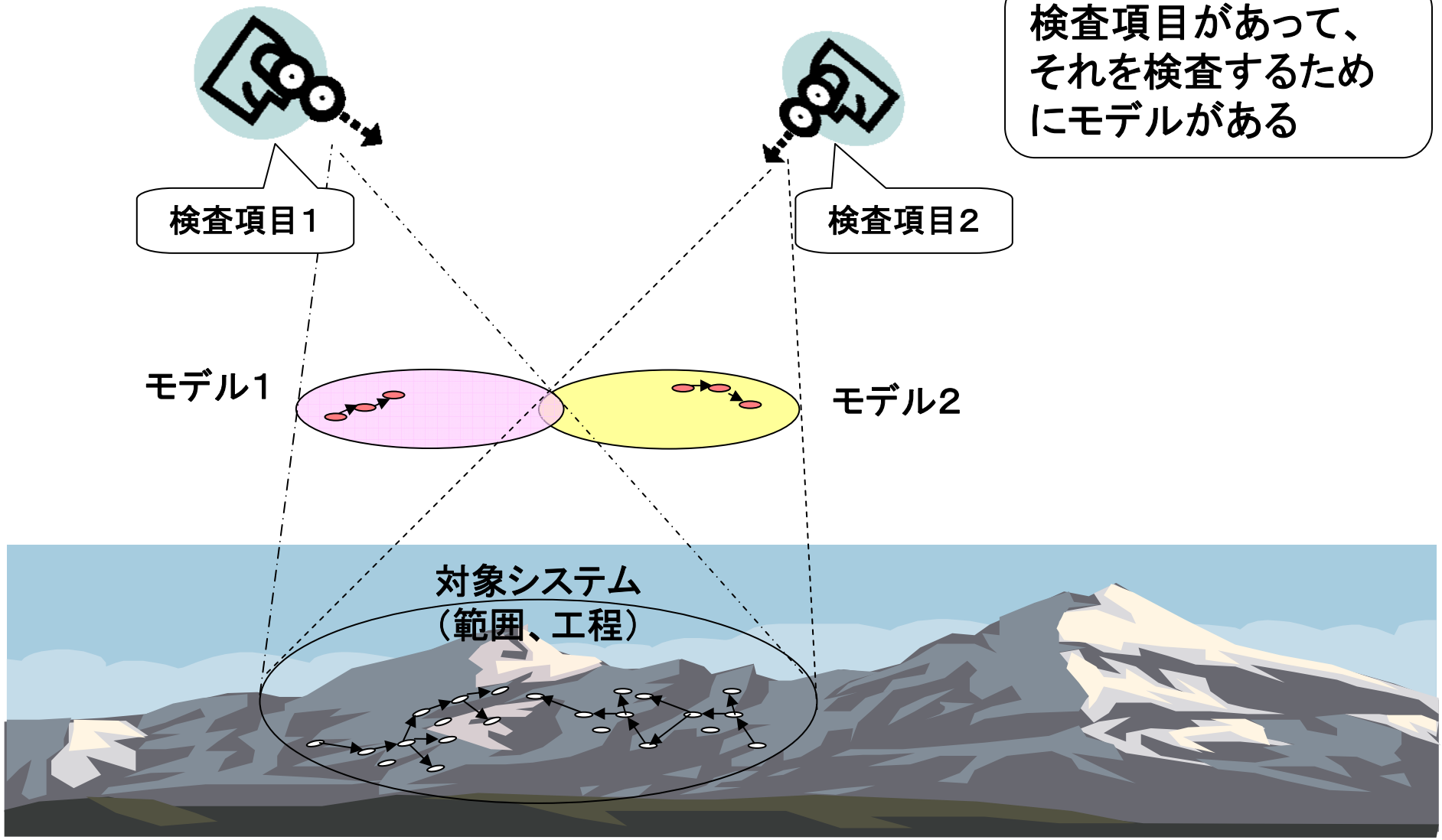
道具の選択（Know your weapons）

モデル検査器

種類	入力言語	特徴
NuSMV	SMV	CTL式の検査 （時制と分岐を扱う）
NuSMV+ （支援ソフトウェア）	フロー図 状態遷移図	モデル化／反例解析が容易
Spin	Promela	LTL式の検査 （時制を扱う）
UPPAAL	状態遷移図	時間制約が扱える

どのモデル検査器も、モデル作成が必要⇔モデルの世界を検査する

モデル化の留意点


検査項目があって、
それを検査するため
にモデルがある
検査項目１
検査項目２
モデル１
モデル２
対象システム
（範囲、工程）

検査項目がモデルを決める

起きてはいけない動作

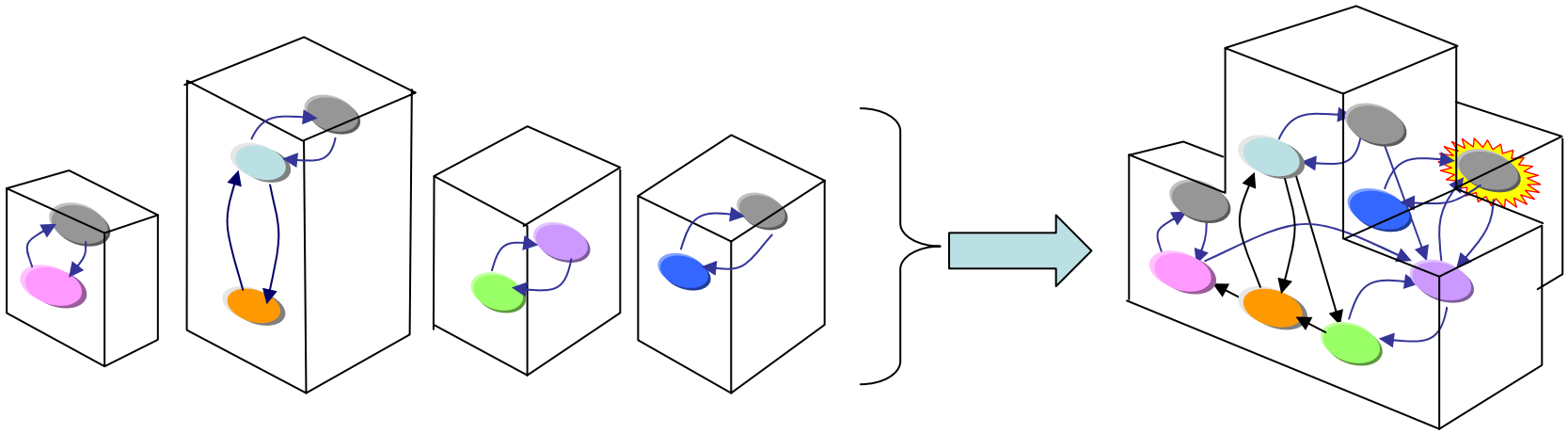
出来ていなければならない動作

→ 動作に必要な要素をモデルに取り込み

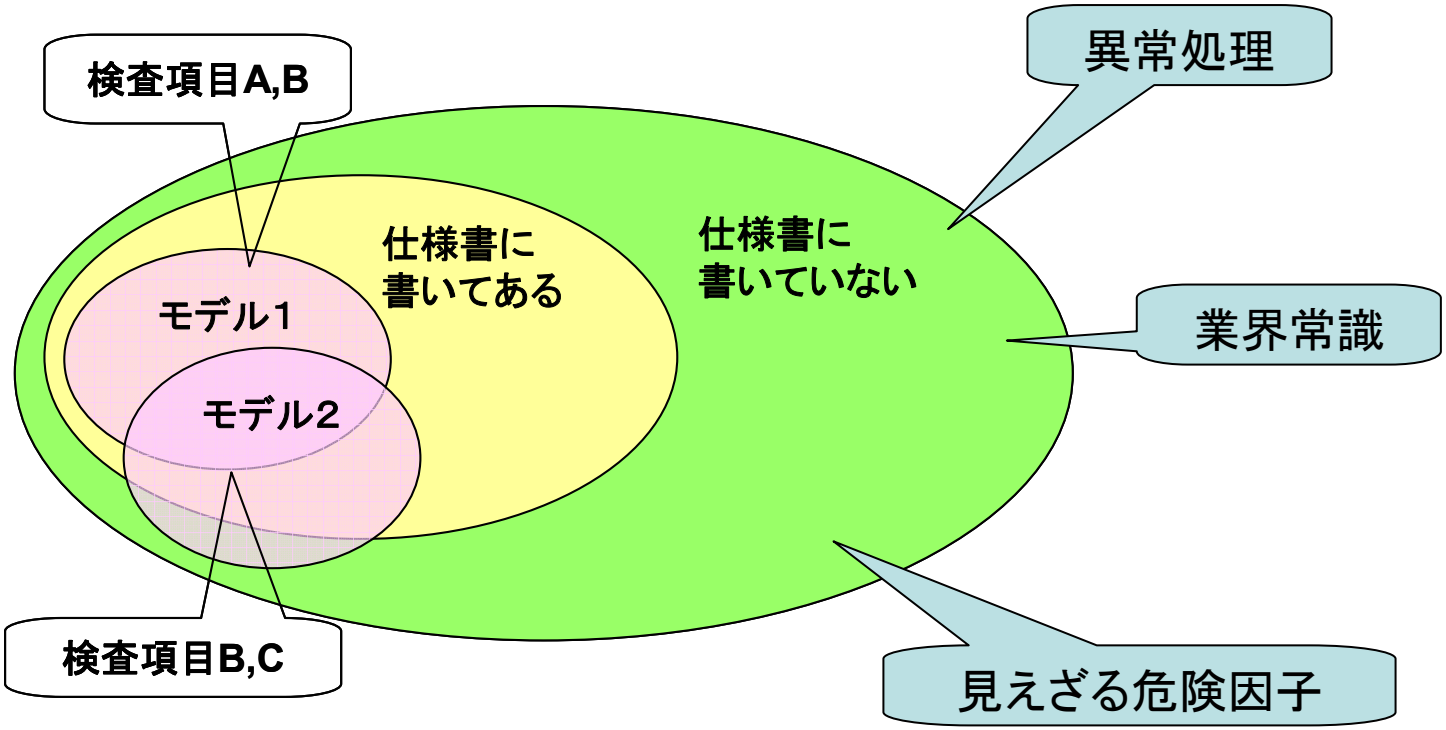
1. 検査したい**要素**がモデルに書かれていること（状態にないものは検査できない）

「どうなるか」は、分からくて良い
「この要素が怪しい」を入れておく

2. システムの部分モデルを作れば良い（全体モデルは自動生成される）



検査項目がモデルを決める

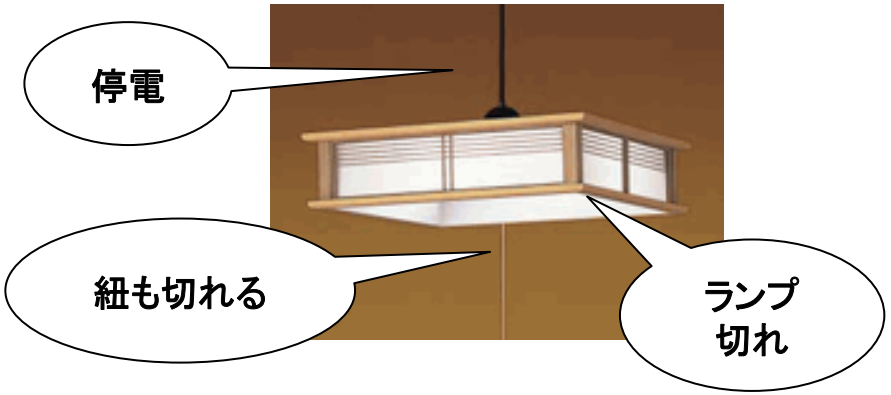


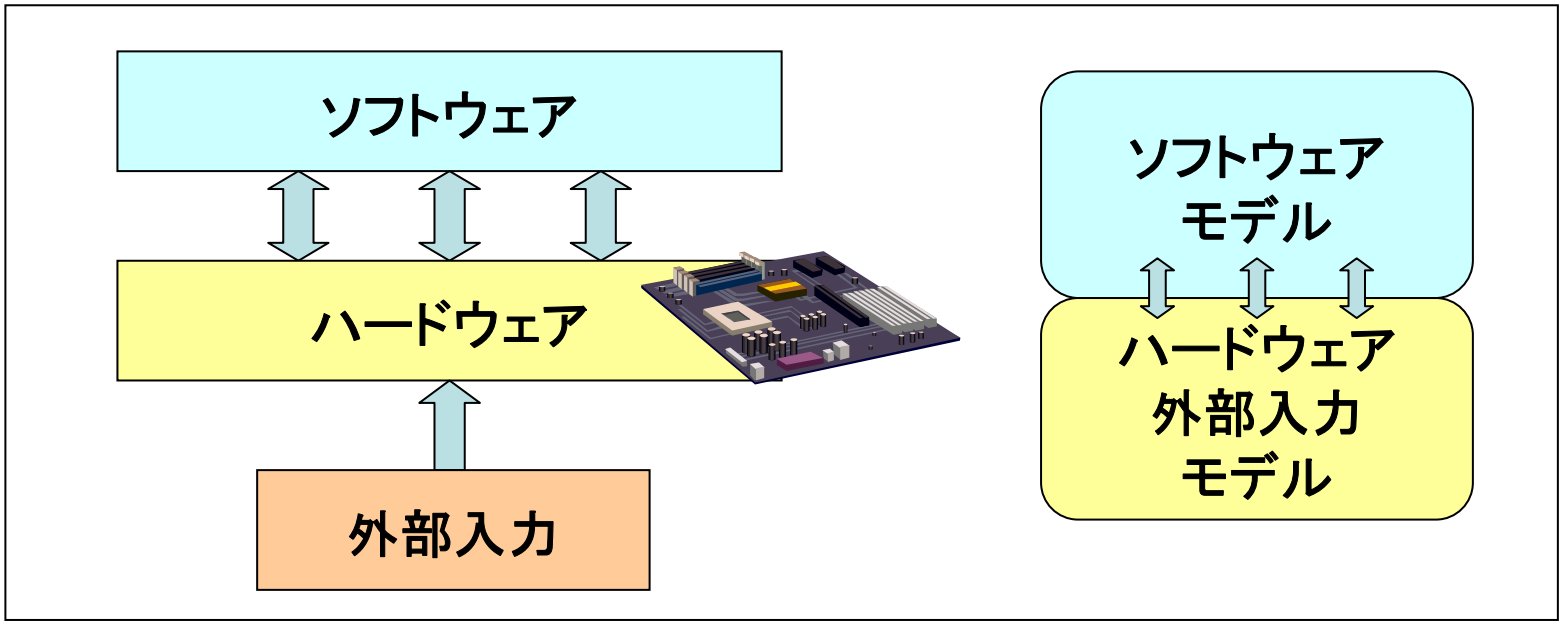
モデルがひとつとは限らないし、仕様書に書いてあるとも限らない

外部環境のモデル

外部環境
（ハードウェア）

外部環境
（外部入力）





外部環境のモデル

最近の事例から

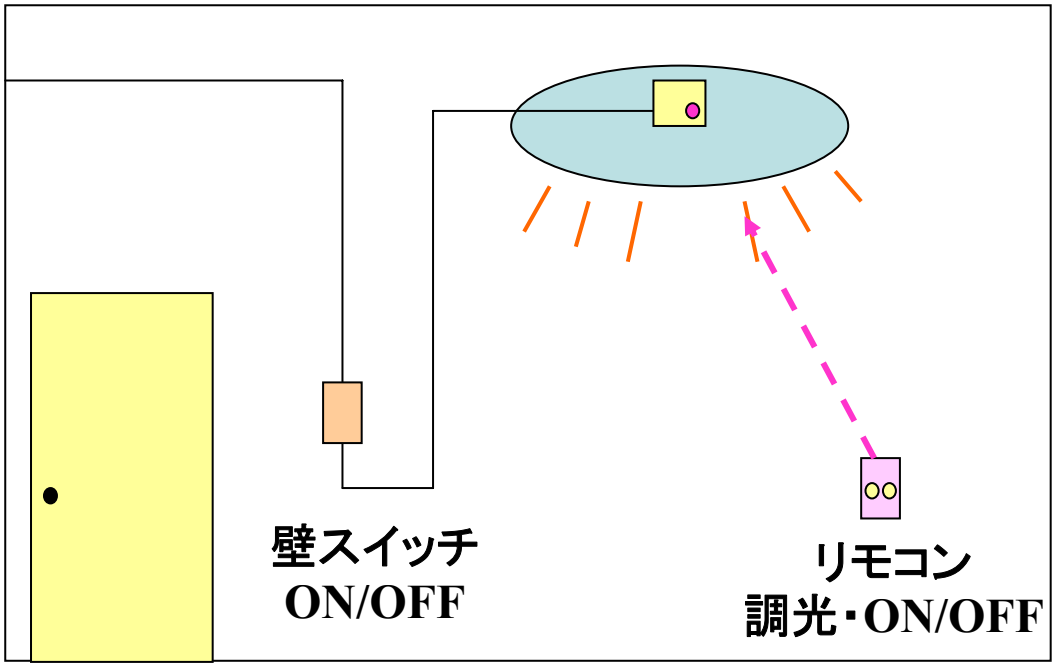
リモコンで消灯

施錠 ・ 外出

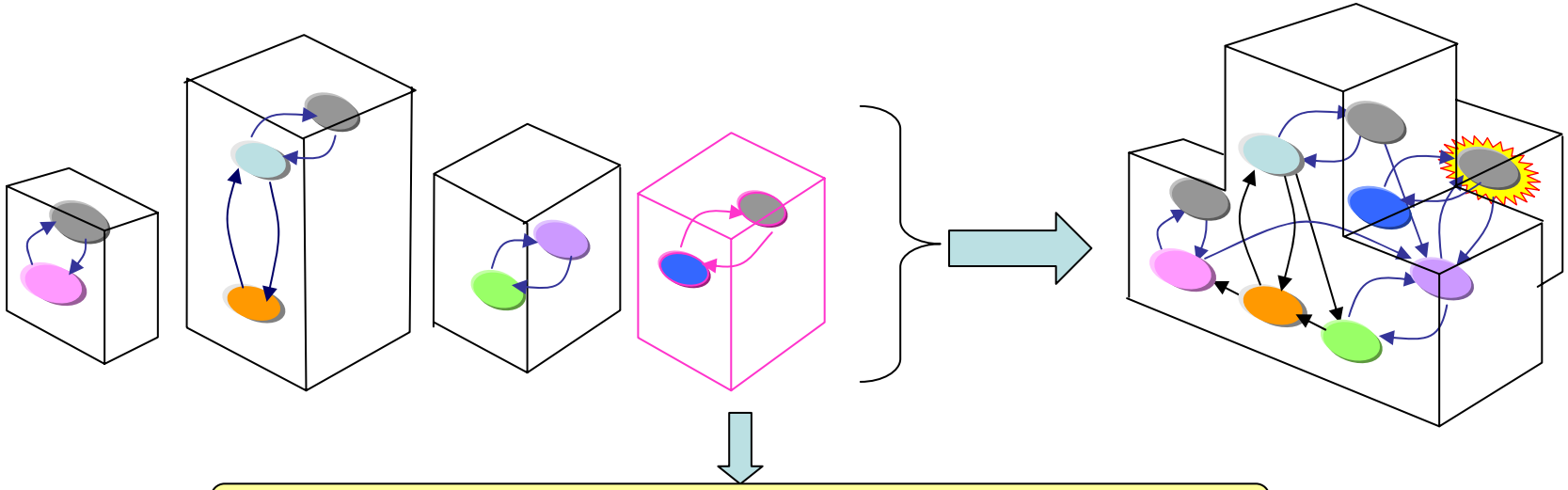
停電 ・ 復帰

壁スイッチ操作と
認識して点灯

外出から帰宅して
びっくり（泥棒？）



外部環境のモデル



外部モデルの必要性はあるが、動作が不明なとき

非決定性を使ってモデル化

- モデル化 ⇒ 容易
- 状態数増 ⇒ 大きい

詳しく調べてからモデル化

- モデル化 ⇒ 難しい
- 状態数増 ⇒ 小さい

検査項目の記述を助けるモデル化

複雑な時相論理式は誤りが発生しやすい

AF(P)　EX(P)
A[P U Q]　E[P U Q]

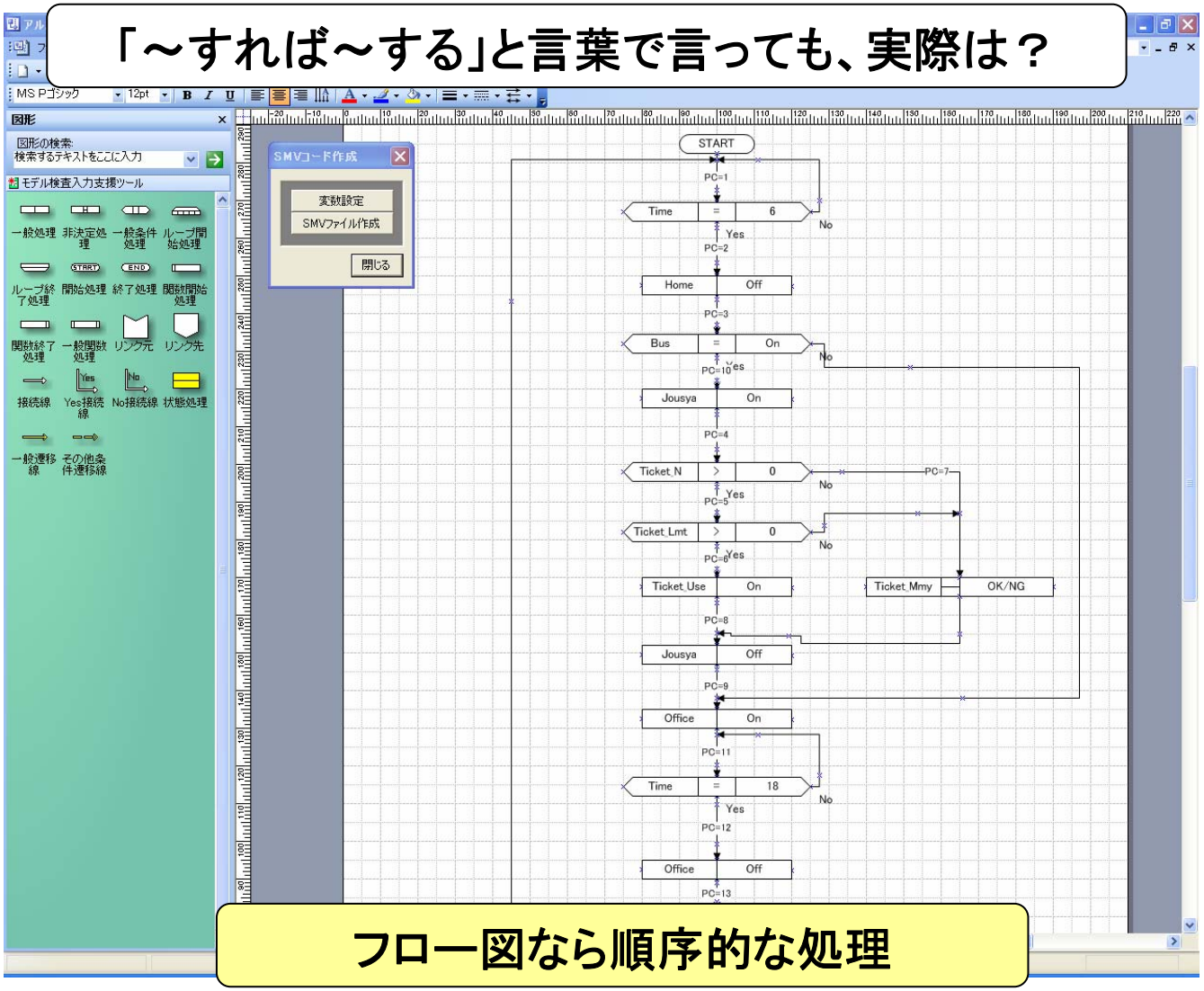
AG((P→AX(!P))&(!P→AX(P)))

SをPがon／offの繰り返しでonになるフラグなら、
AG(S)

モデルに予め検査用のモデル（フラグ）を組み込んでおく。⇒式はシンプルに。

入出力のタイミング

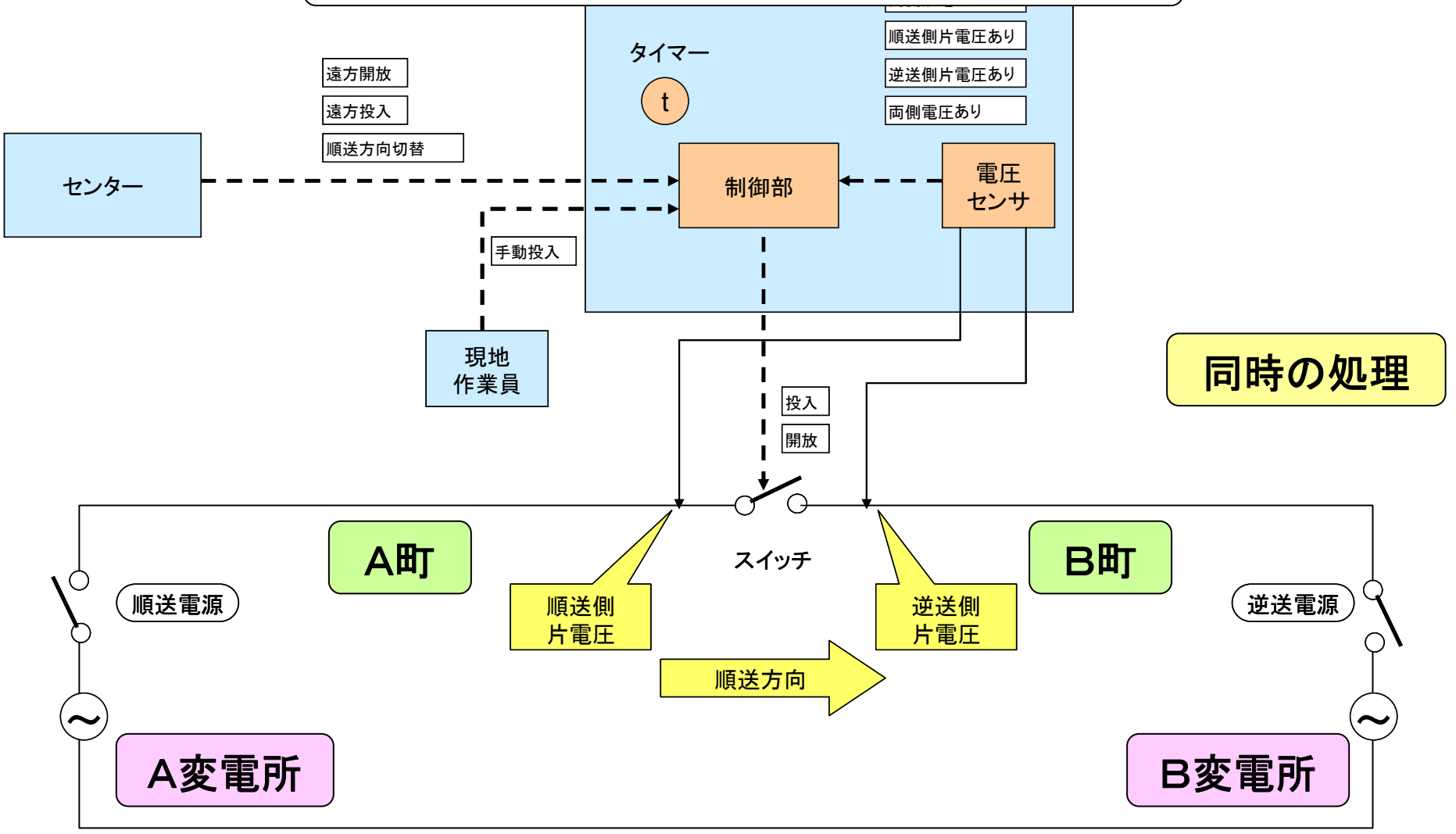
「～すれば～する」と言葉で言っても、実際は？



フロー図なら順序的な処理

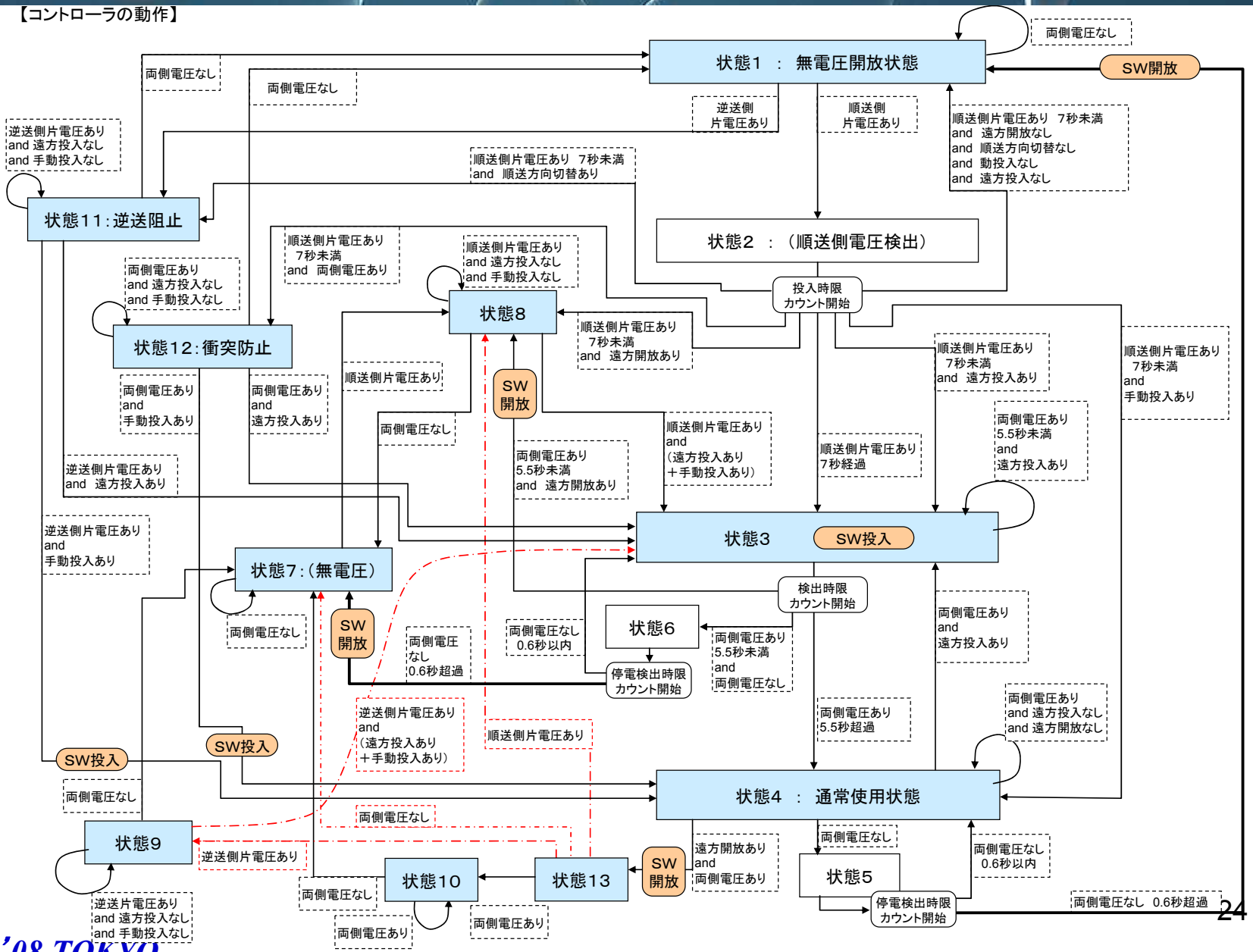
入出力のタイミング

「～すれば～する」と言葉で言っても、実際は？



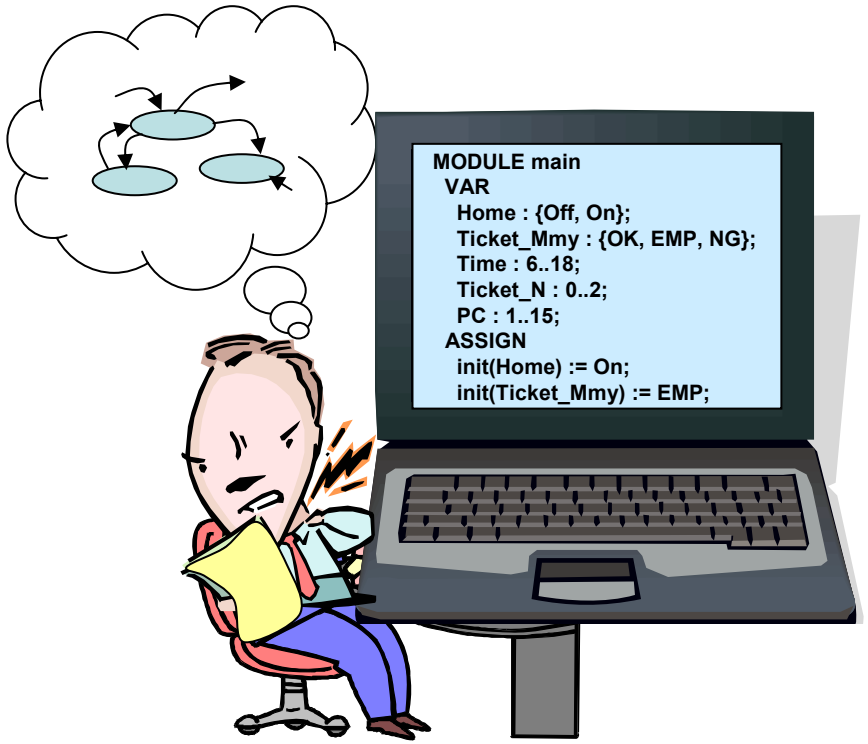
同時処理の例（遷移条件で同時に判断）

【コントローラの動作】



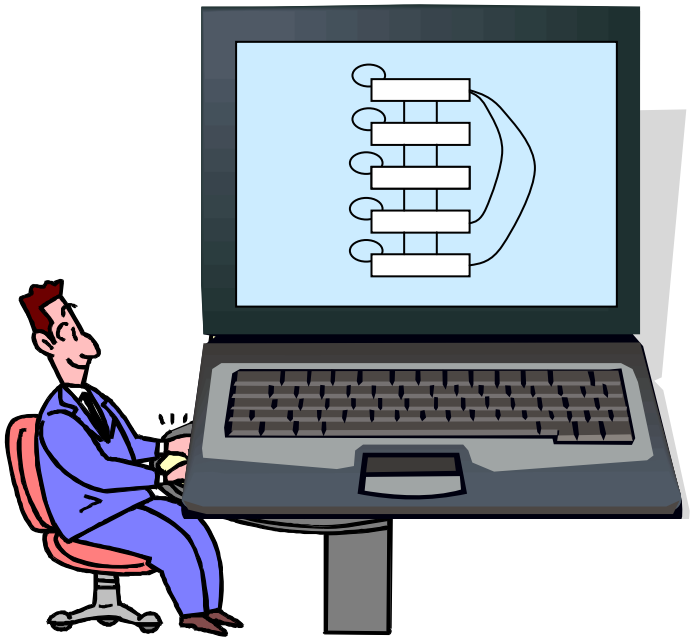
モデル検査の実施

従来のモデル検査



・記述力は高いが学習、経験が必要
・作業時間が長い

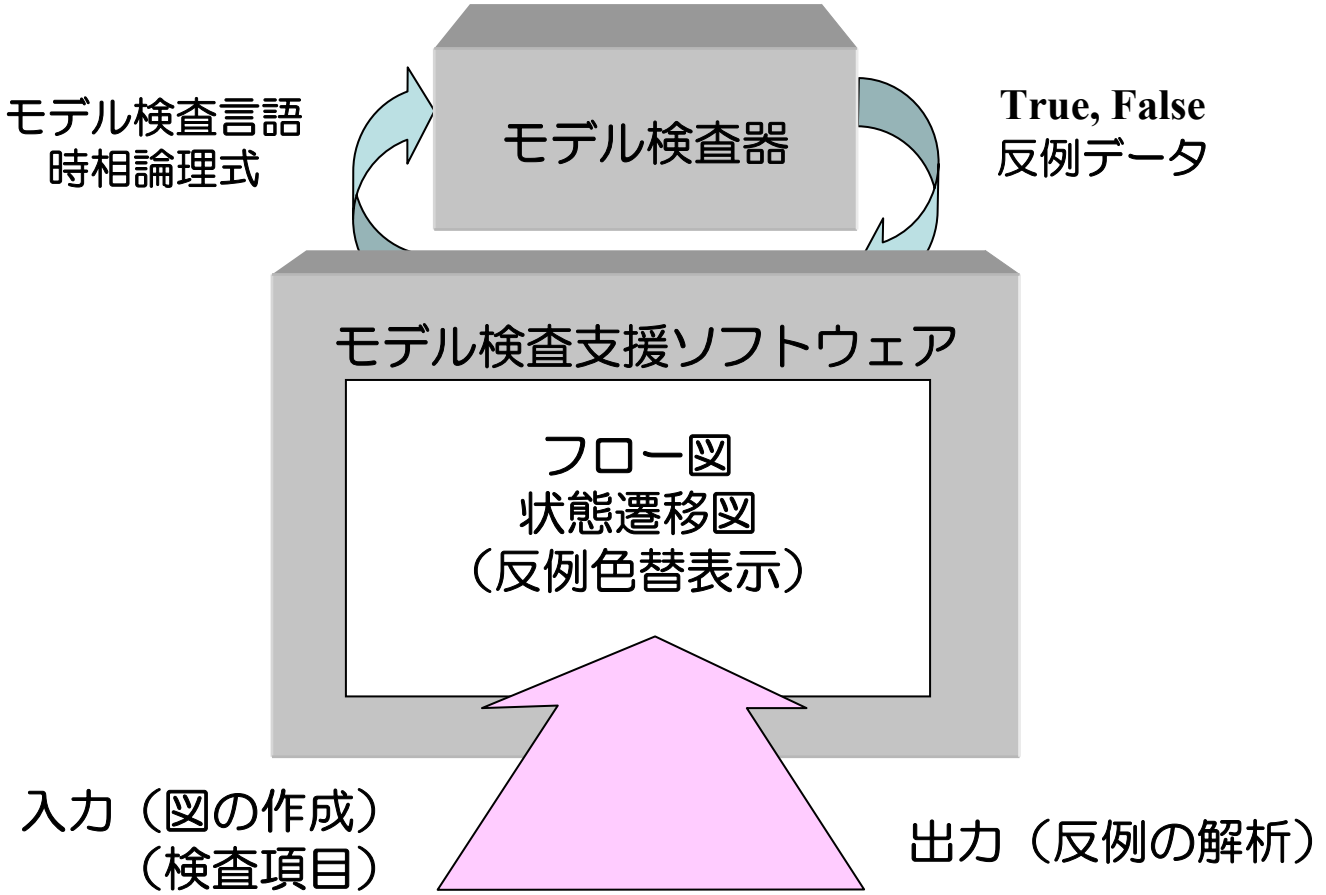
モデル検査支援ソフトウェア



・学習、経験が短時間で済む
・作業時間を1/5に短縮

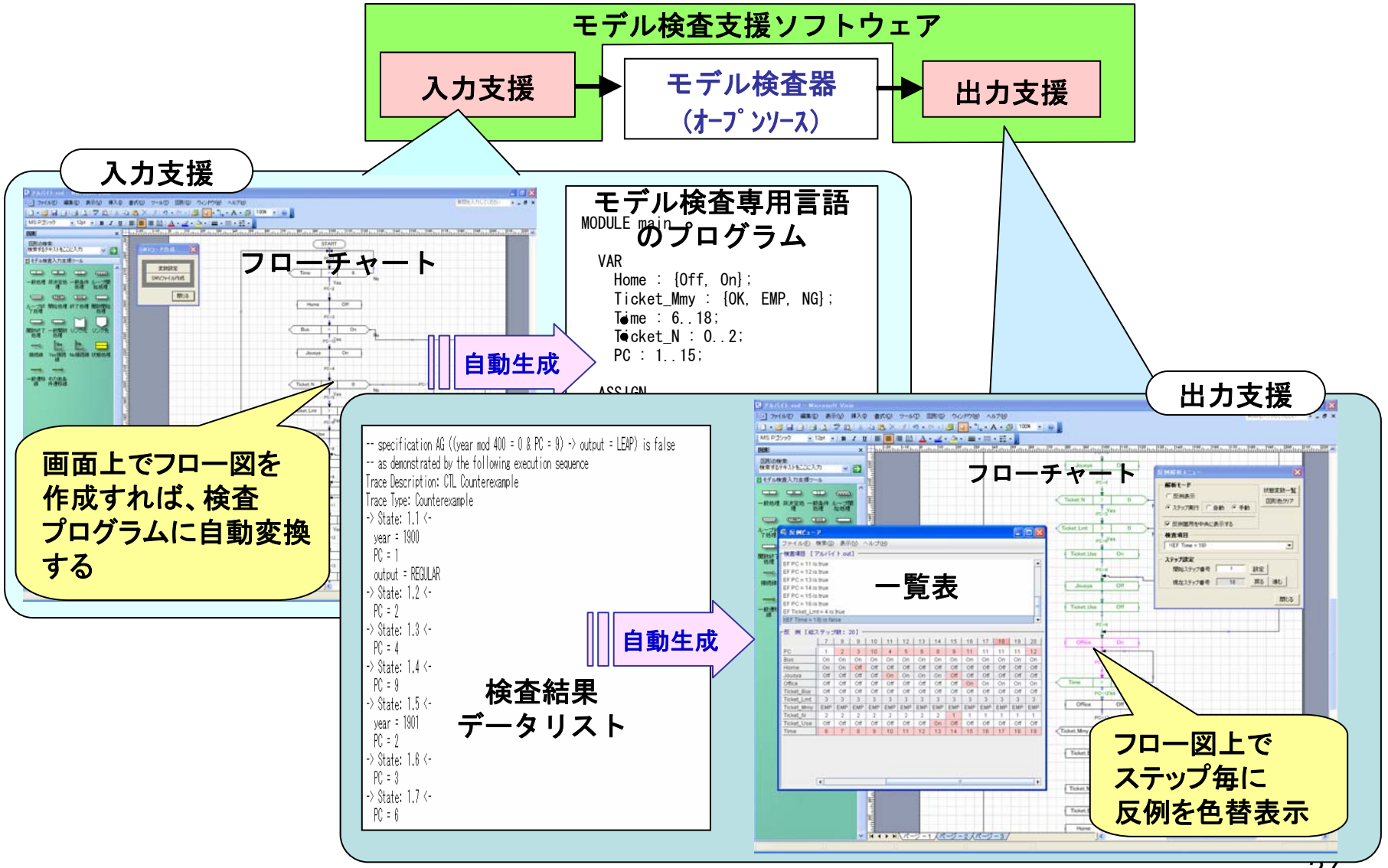
検査する道具⇒動作を理解する道具⇒遷移図を正しく書く道具

モデル検査支援ソフトウェア

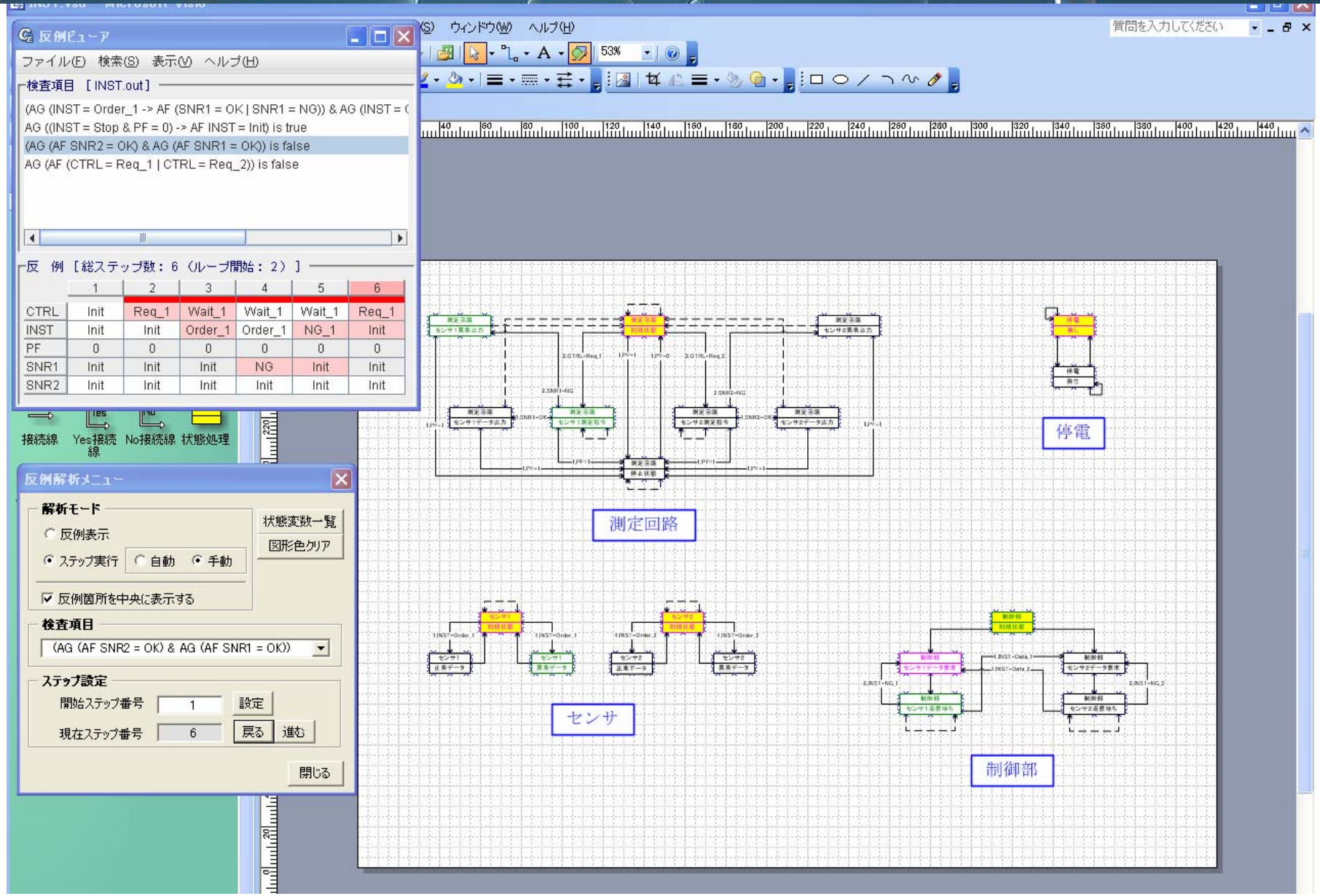


試供版CD-R配布します

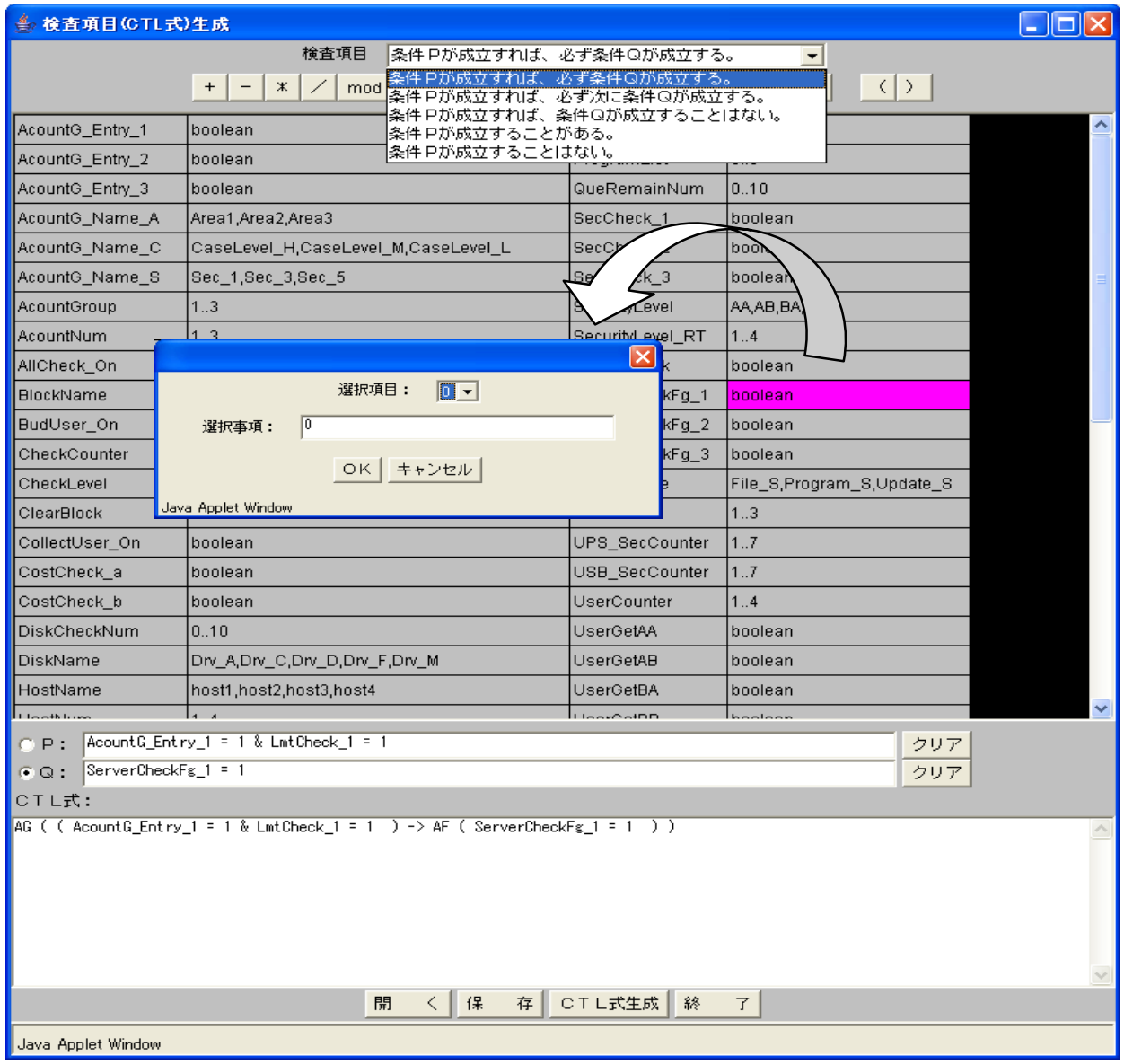
モデル検査支援ソフトウェア



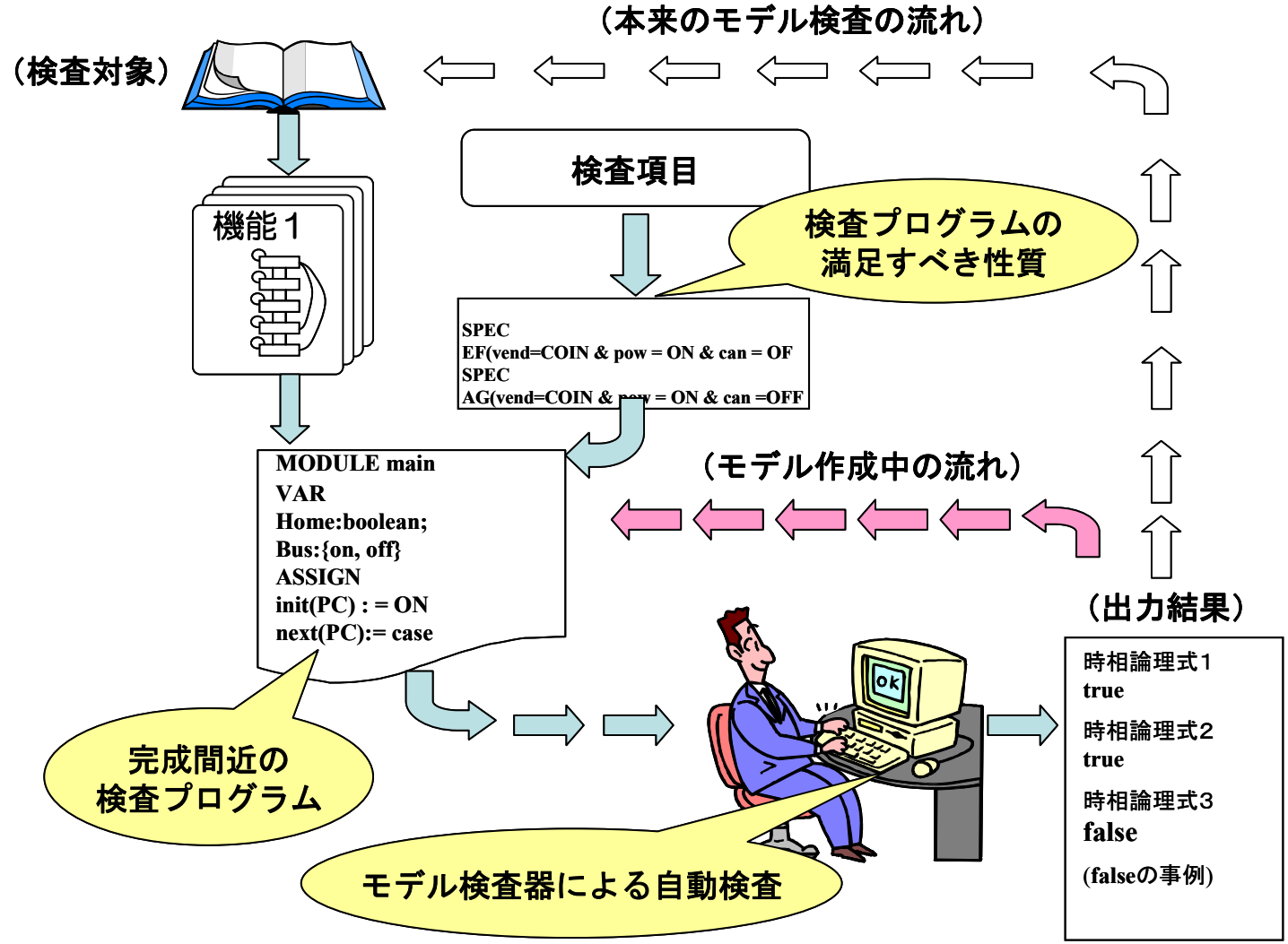
モデル検査支援ソフトウェア



モデル検査支援ソフトウェア



モデル作成途中でのモデル検査

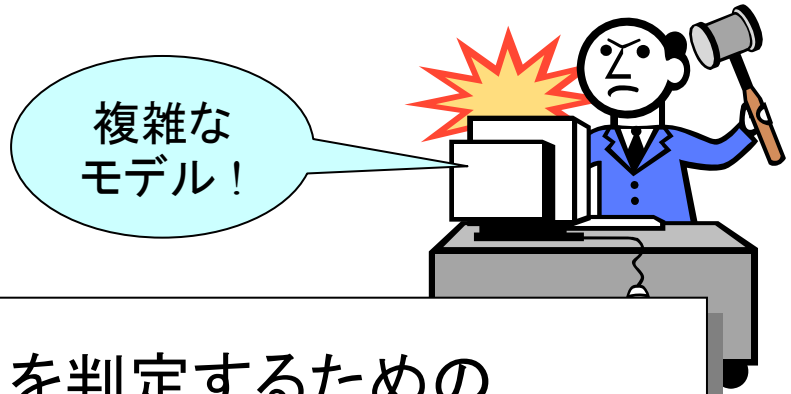


モデル検査の作業ループを回すのが楽になった

小さなモデルから検査開始

課題

状態爆発による無回答

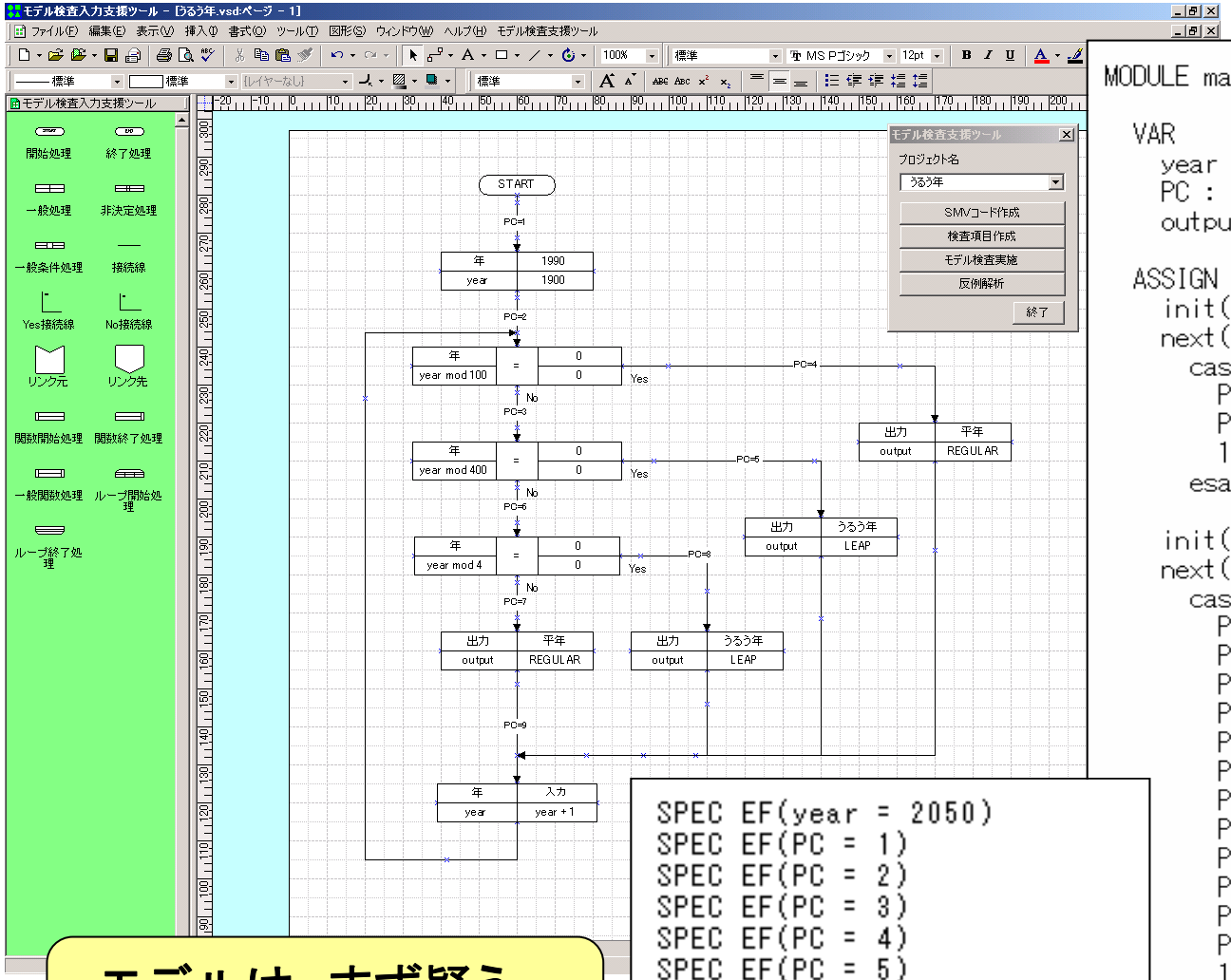


対策

実行継続・中止を判定するための
閾値を求める

単純なモデル・検査項目から出来具合も見ながら複雑に

モデルの事前確認




```
MODULE main

  VAR
    year : 1900..2050;
    PC : 1..9;
    output : {LEAP, REGULAR};

  ASSIGN
    init(year) := 1900;
    next(year) :=
      case
        PC = 1 : 1900;
        PC = 9 : {1900..2050};
        1 : year;
      esac;

    init(PC) := 1;
    next(PC) :=
      case
        PC = 1 : 2;
        PC = 2 & year mod 100 = 0 : 4;
        PC = 2 & !(year mod 100 = 0) : 3;
        PC = 3 & year mod 400 = 0 : 5;
        PC = 3 & !(year mod 400 = 0) : 6;
        PC = 4 : 9;
        PC = 5 : 9;
        PC = 6 & year mod 4 = 0 : 8;
        PC = 6 & !(year mod 4 = 0) : 7;
        PC = 7 : 9;
        PC = 8 : 9;
        PC = 9 : 2;
        1 : PC;
      esac;
```

```
SPEC EF(year = 2050)
SPEC EF(PC = 1)
SPEC EF(PC = 2)
SPEC EF(PC = 3)
SPEC EF(PC = 4)
SPEC EF(PC = 5)
SPEC EF(PC = 6)
SPEC EF(PC = 7)
SPEC EF(PC = 8)
SPEC EF(PC = 9)
```

- モデルは、まず疑う
- 基本的な性質を検査

検査項目の事前確認

AG(FRE = 1 → AX(FDL=1))

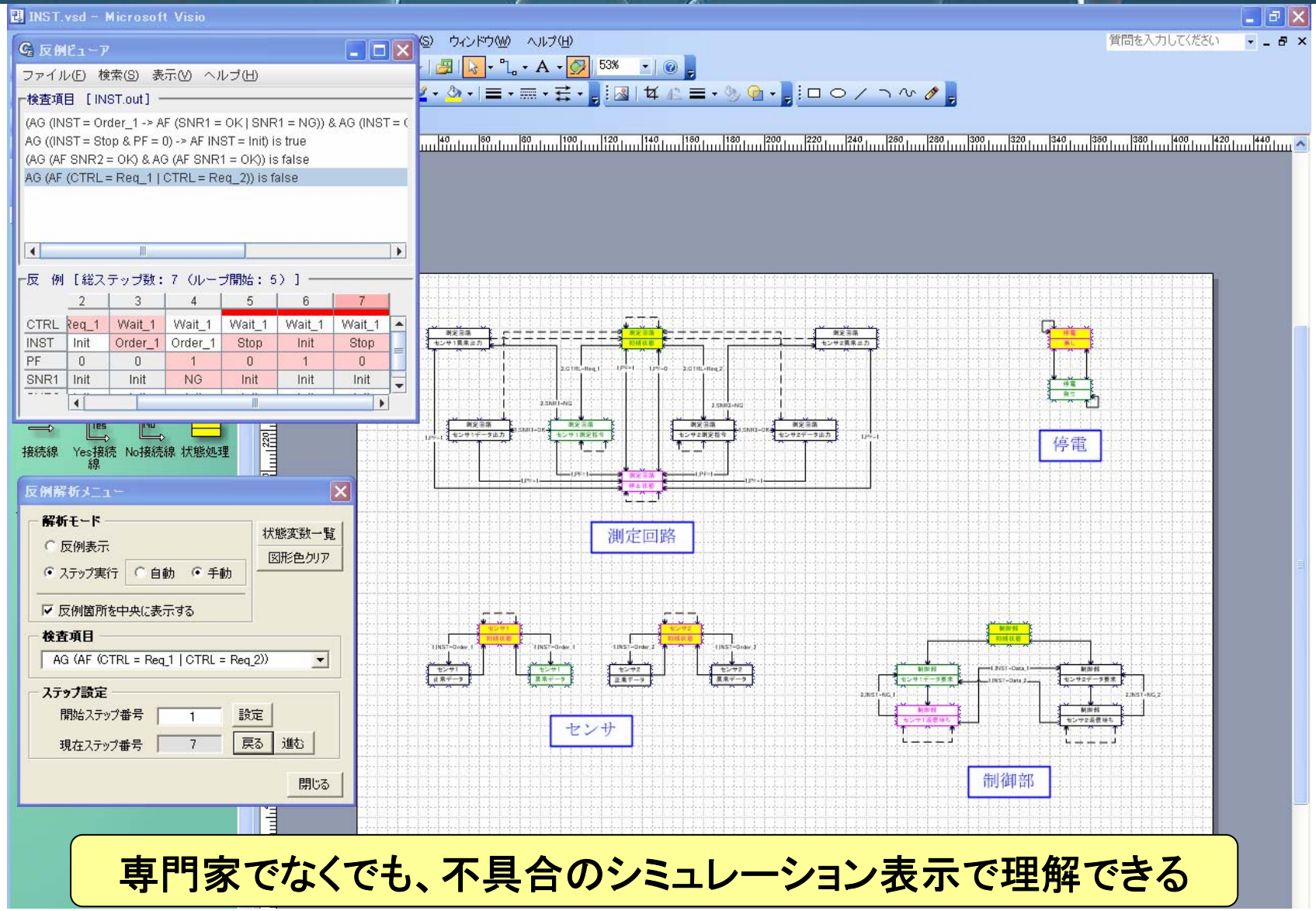
日本語の「～ならば」に
近い意味（厳密には違う）

前提条件が成立しないと「True」になる

EF(FRE = 1)

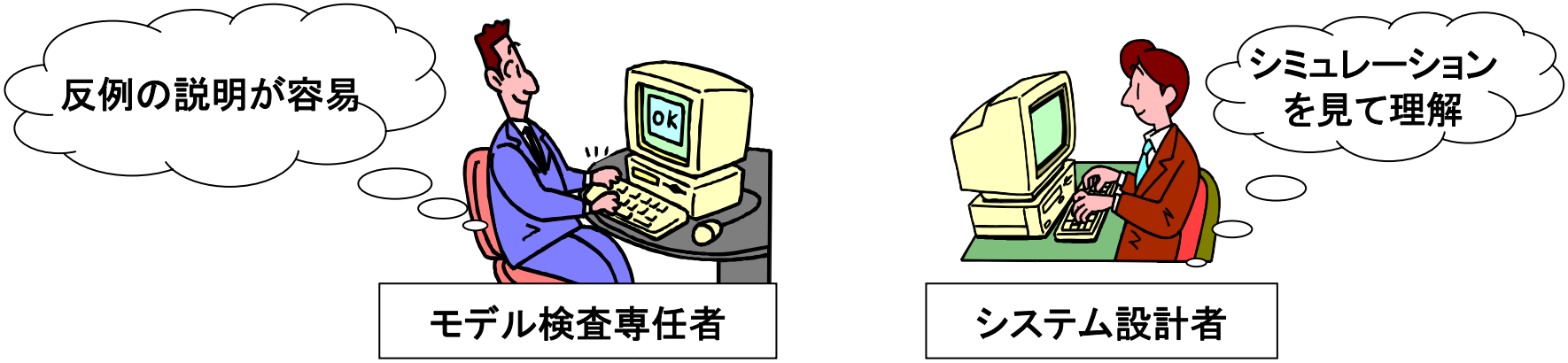
前提条件の成立を先に検査しておく

反例解析での支援ソフトウェア活用



専門家でなくでも、不具合のシミュレーション表示で理解できる

反例の評価



発生頻度

発生事象の重大さ

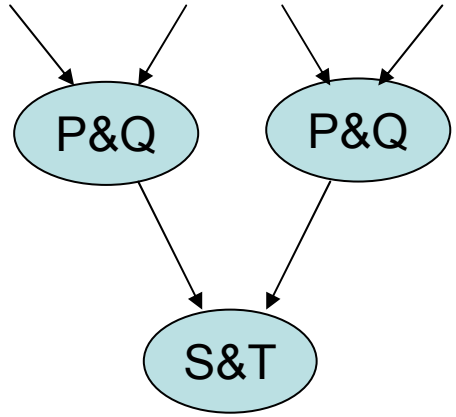
対策要否の判定

「対策なし」でもリスク明確化

反例の変更

!EF(P&Q)

モデルを修正しても同じ反例ばかり出る

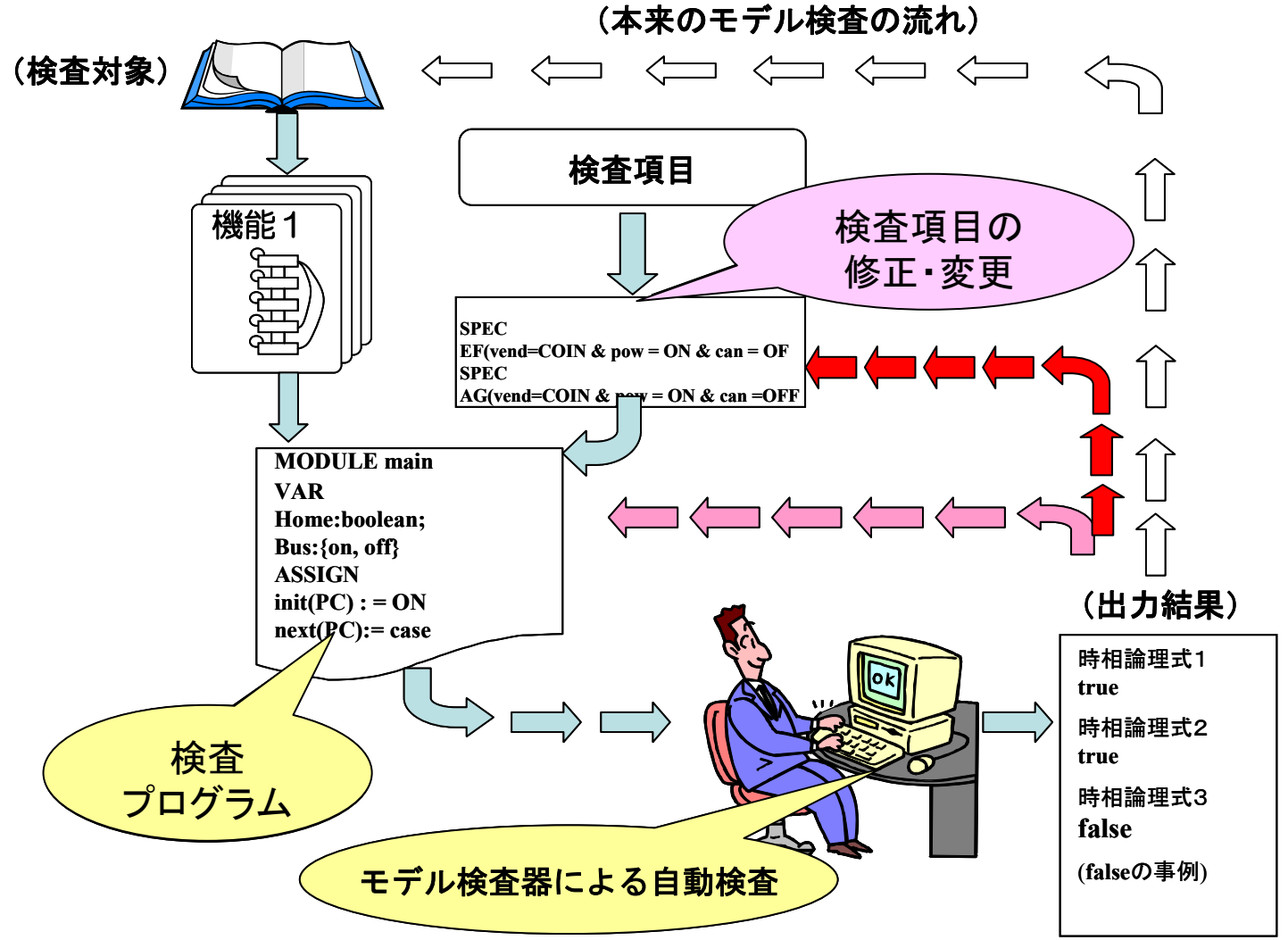


検査項目を変更してみる

!EF(P | Q)　　!EF(P＆Q＆R)　　!EF(S＆T)

検査項目を（満足すべきものだけでなく）
システムの動作を調べる道具と考える

何度でも気楽にモデルを回す



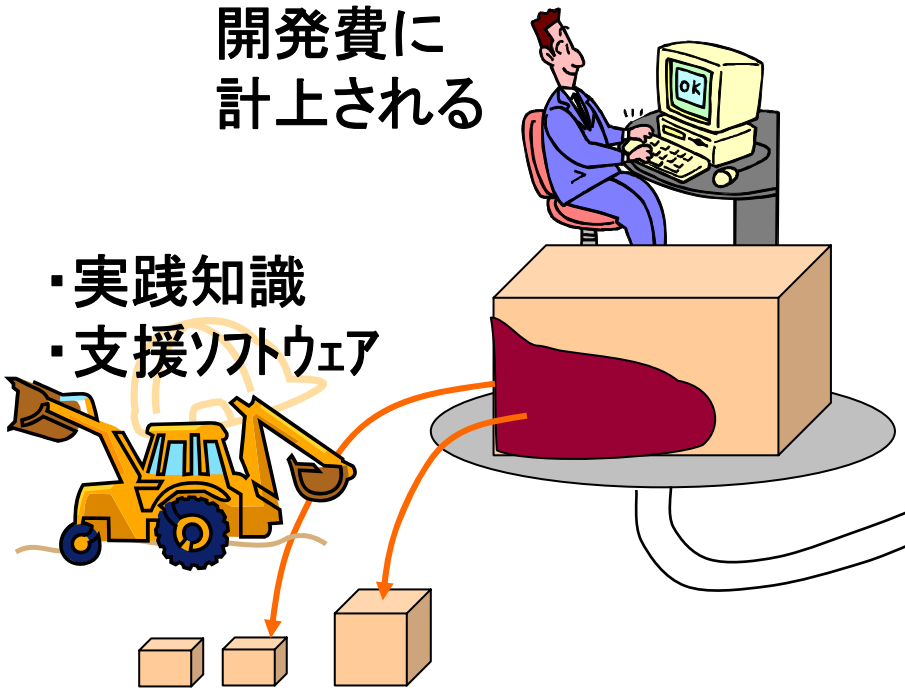
検査項目を変更しながらシステムの不具合動作を理解する

モデル検査の実践に向けて

モデル検査に要した時間

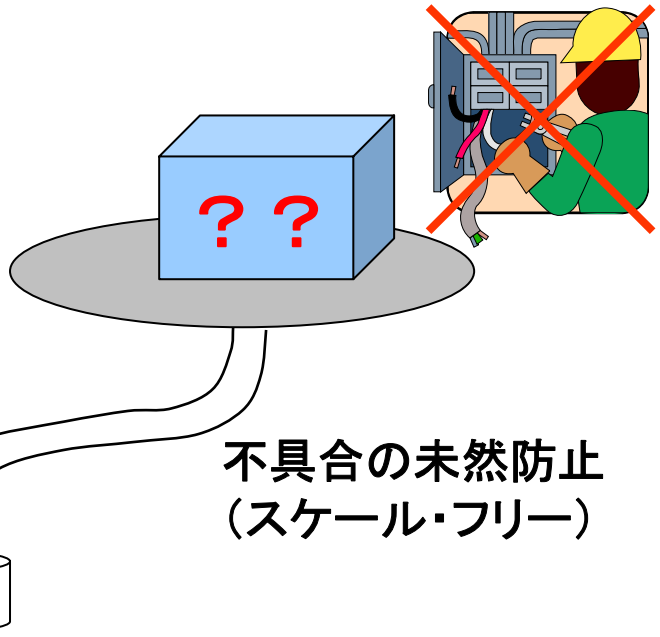
開発費に
計上される

・実践知識
・支援ソフトウェア



試行錯誤の時間

モデル検査の時間効果

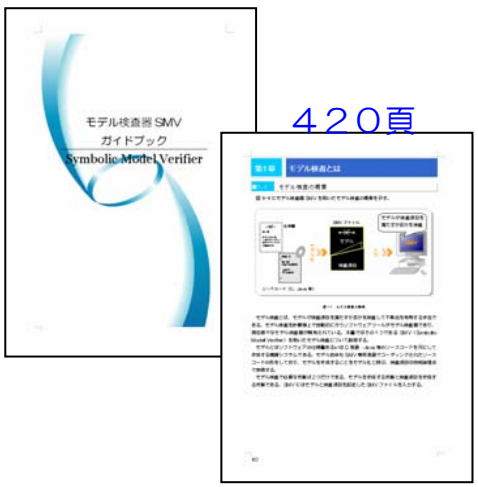


不具合の未然防止
（スケール・フリー）

様々な実践知識を活用して、効率的なモデル検査実施が必要

実践知識のとりまとめ

実践知識の取りまとめ



４２０頁

ガイドブック

テキスト

支援ソフトウェアを使った
モデル検査チュートリアル

企業内での実践教育

ご清聴ありがとうございました

http://www.modelcheck.jp/　にて
モデル検査支援ソフトウェア試供版提供中

関西電力株式会社
メルコ･パワー･システムズ株式会社