ModelChecking技術の専門性の排除とその効用

An EarlyBird as the 2nd eye

富士ゼロックス株式会社
オフィスプロダクト事業本部
コントローラソフトウェア開発部

野村 秀樹/服部　彰宏/村石 理恵/山本 訓稔

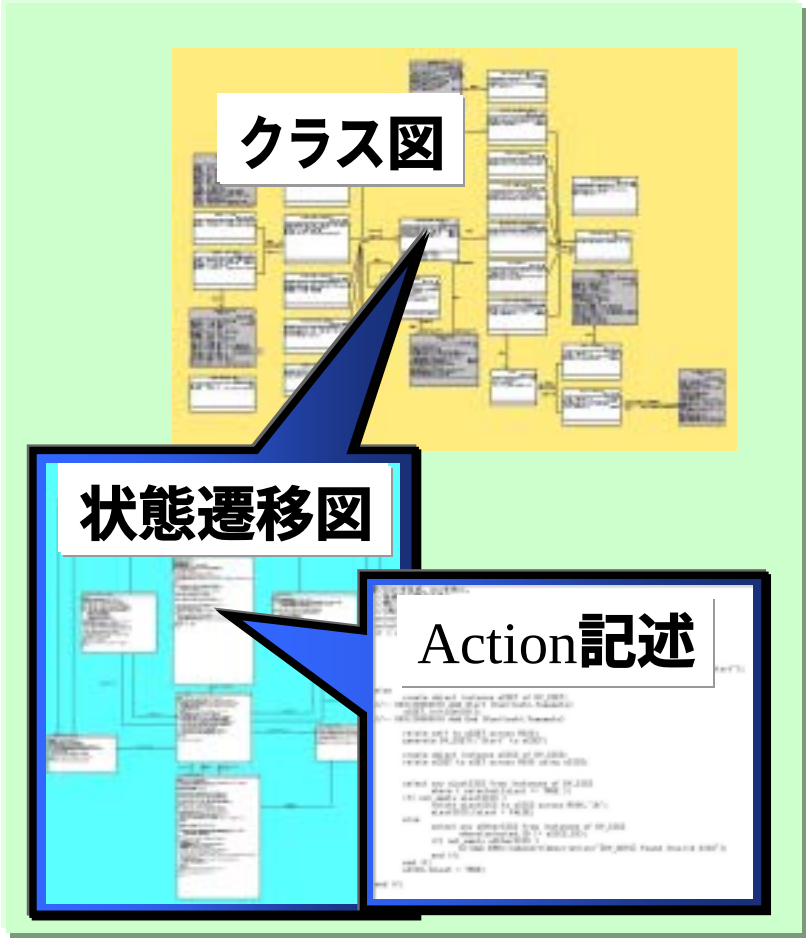
2008 年 1月30日

1. 我々のMDDの実践状況

THE DOCUMENT COMPANY
FUJI XEROX

Model

・**シミュレーション実行可能** (Executable)
・**モデルがレビュー対象**



C言語への自動変換結果

・**製品そのもの (完全自動変換)**
・**コードはレビュー対象外**

```
/* Set Close */
void
IOTim_dev_DV_SETC_Action_1( IOTim_dev_DV_SETC_s * self, const OoaEvent_t
* const event )
{
  IOTim_dev_DV_OUTS_s * v458;  /* aOUTS */
  IOTim_dev_DV_OUT_s * v461;  /* aRegistingOUT */
  IOTim_dev_DV_OEOS_s * v464;  /* aOEOS */
  IOTim_dev_DV_OEOJ_s * v467;  /* aOEOJ */
  IOTim_dev_DV_OSOS_s * v470;  /* aOSOS */
  IOTim_dev_DV_SHT_s * v473;  /* aSHT */
  IOTim_dev_DV_SHT_s * v477;  /* tmpSHT */
  IOTim_dev_DV_CSET_s * v483;  /* aCSET */

  /* SELECT ANY aOUTS FROM INSTANCES OF DV_OUTS */
  v458 = (IOTim_dev_DV_OUTS_s
*)Escher_SetGetAny( pG_IOTim_dev_DV_OUTS_extent );

  /* SELECT ONE aRegistingOUT RELATED BY aOUTS->DV_OUT[R311] */
  v461 = v458->mc_DV_OUT_R311;

  /* IF (NOT_EMPTY aRegistingOUT
  if ( (((v461 != 0)
  {
*/ /* S                                    gistingOUT->DV_OEOS[R304]
```

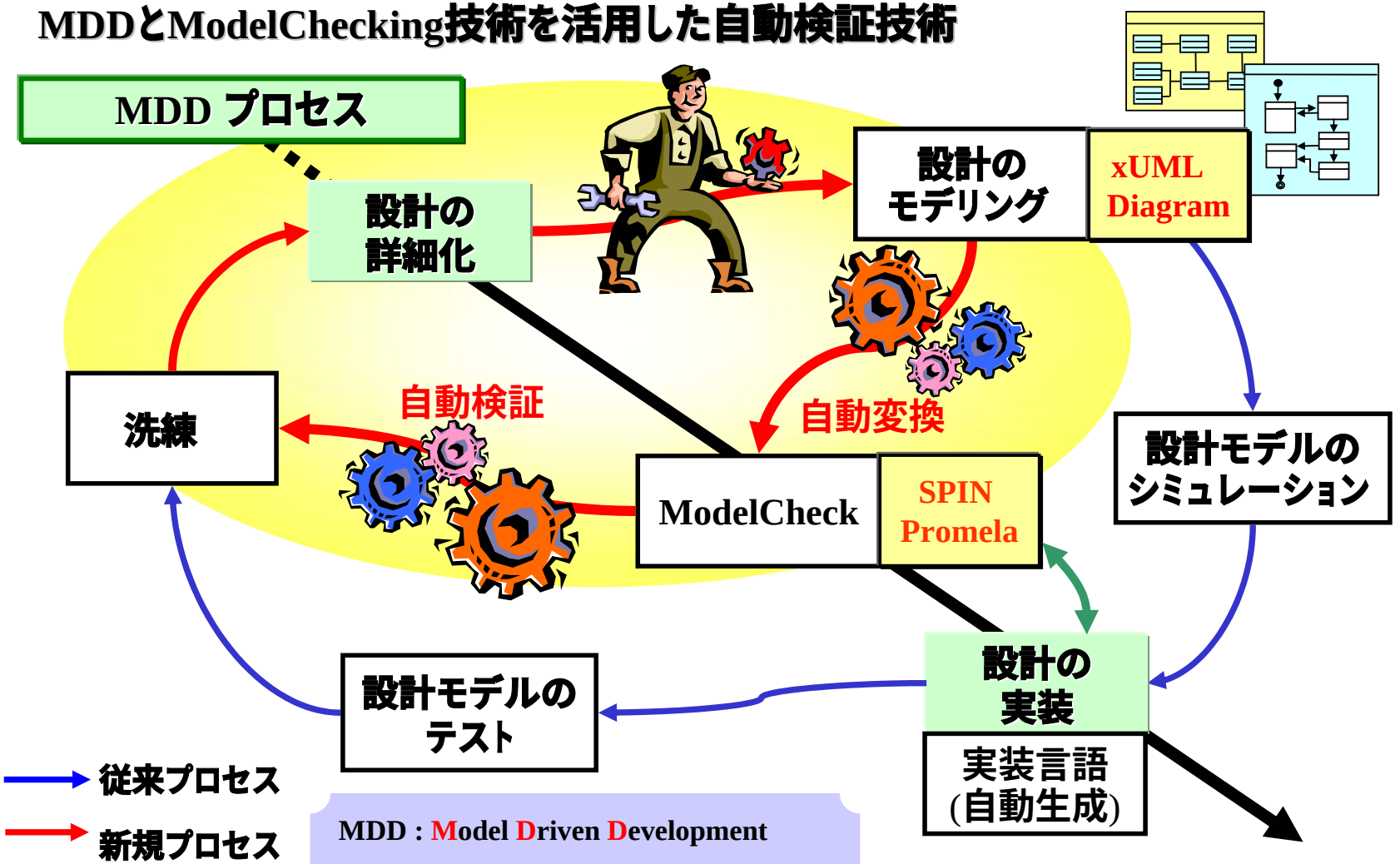
スケルトンではない

・・・後略・・・

2. EarlyBirdプロセス

THE DOCUMENT COMPANY
FUJI XEROX

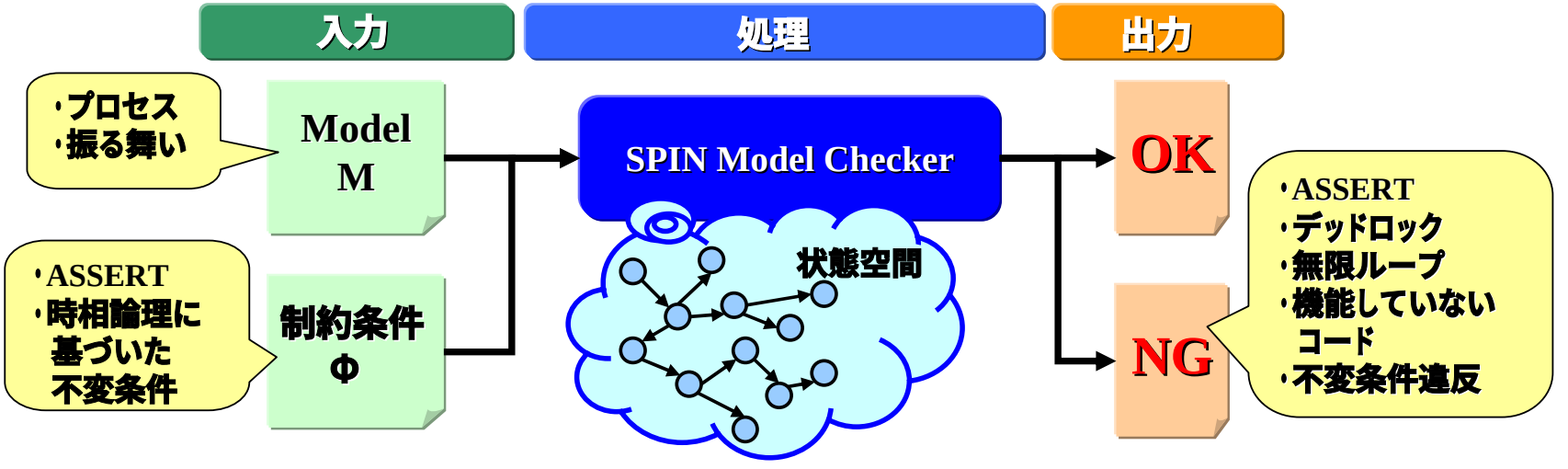
MDDとModelChecking技術を活用した自動検証技術



3. ModelCheckingとは？

THE DOCUMENT COMPANY FUJI XEROX

システムをモデル化して、その状態を全てシミュレーションし、仕様の正しさを抜け漏れなく検証する技術



- SPIN (=Simple Promela Interpreter)　開発者:G.J.Holzmann
  - Promelaで書かれたモデルの検証を行うツール
  - 検証対象のシステムの状態空間を制約条件に基づいて徹底的に検証する
  - NGの箇所で停止し、NGまでの到達経路をLogとして保存する
- Promela(=Protocol / Process Meta Language)
  - モデル記述言語
  - 非同期通信に関する検証が可能
  - ネットワーク・プロトコル、電話交換機システムなどに適用

THE DOCUMENT COMPANY
FUJI XEROX

DM (Device Mgr.)

■概略

DeviceManagerは、メカ側で発生するイベントに関して
サービス制御用コントローラ側の他モジュールとの調停を行う。

■開発手法

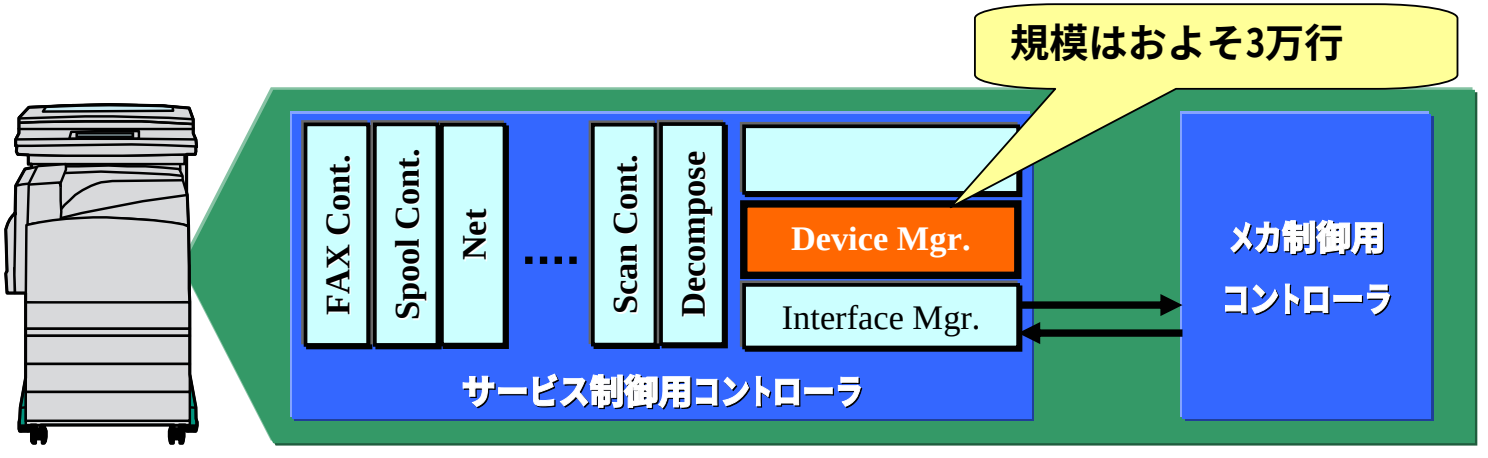
MDD

■開発規模

３０Kloc程度の規模

■開発担当者

SPIN/Promelaの知識を身につけていない。



5. やらなければならなかった事

THE DOCUMENT COMPANY
FUJI XEROX

■置かれていた状況

開発終盤で設計変更が必要な不具合が発生

■要求される達成レベル

・再設計時に考慮漏れがあってはならない
・2次不具合は出してはいけない

・従来の実機テストでの確認で十分??
・二次不具合は本当にない??

不安・・・

ModelCheckingで網羅テストすれば
二次不具合がないことを確認できそう！！

安心!!

6. 今回やりたかったこと

THE DOCUMENT COMPANY
FUJI XEROX

手段であって
目的ではない

おいしいところだけ

What

・不具合の**原因検証**と
対策の**十分性の検証**

How

・M/Cの**勉強はしない！**

・M/Cはただの**道具** ♪

7. 必要なこと

THE DOCUMENT COMPANY
FUJI XEROX

今回やった事

Step 1. 使えるようにする

不具合対策の検証に専念できるようにしたい

↓

検証モデル作成プロセスからSPIN固有の
専門性を排除する

Step 2. 実際に使ってみる

本当にやりたいことは不具合対策の検証

↓

不具合“A”対策の検証にModel Checkingを
適用する

8. やった事1

THE DOCUMENT COMPANY
FUJI XEROX

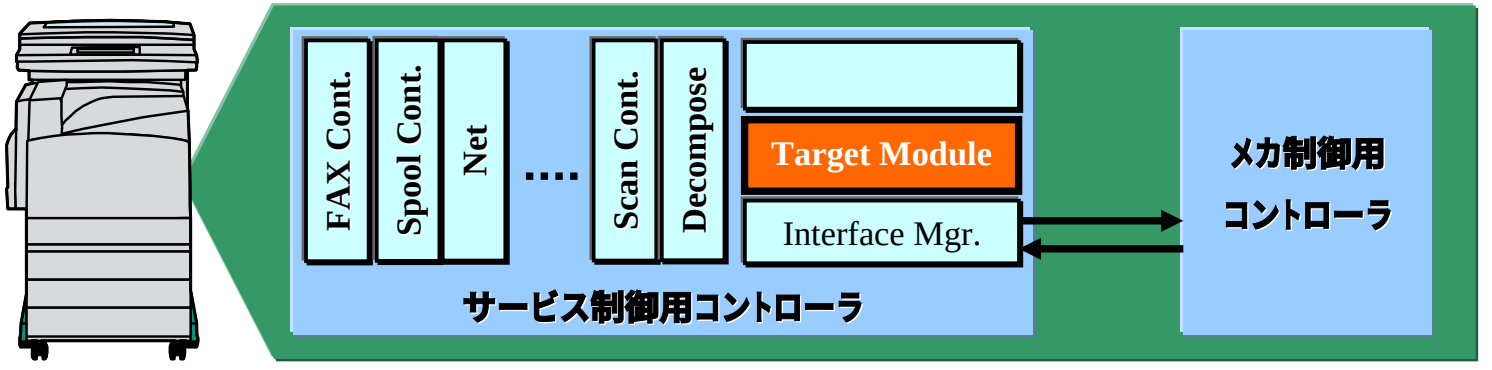
やった事　1　：　専門性の排除

Promelaを知らなくても、外部モデル構築上の
専門知識を知らなくても、Model Checkができるようにする

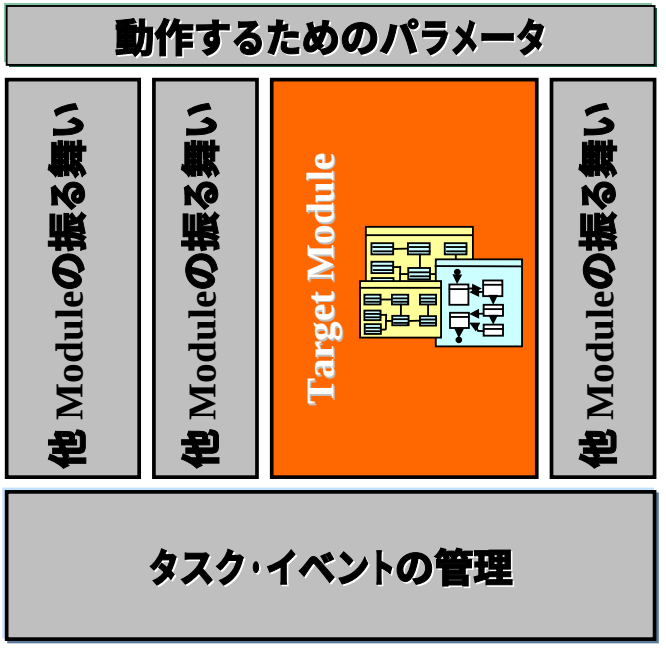
9. 検証モデルの構成

THE DOCUMENT COMPANY FUJI XEROX

製品の構成



検証モデルの構成



検証モデル

- **内部モデル･･･抽象化しない**
  - Target Module
- **外部モデル･･･抽象化する**
  - 他Moduleの振る舞い
  - 動作するためのパラメータ
  - タスク･イベントの管理

※抽象化とは･･･
検証目的に合わせて、モジュールの振る舞いに制限を加え、検証空間のサイズを小さくすること。

10. 外部モデル作成に要求される専門性

THE DOCUMENT COMPANY FUJI XEROX

外部モデル作成の専門性

■タスク･イベントの管理

内部･外部モデルが保有するタスク間でやり取りされるイベントの振り分けと優先順位の管理を行う。
SPINの特質に合わせ実機と全く同じ順序で処理されるように設計する必要がある

■他モジュールの振る舞い

内部モデル以外のシステム構成要素を必要な範囲で抽象化し、必要最小限の非同期タスクを用意する。
他モジュールの振る舞いに関する深い知見とSPINの特質に合わせたタスク設計の知見が必要

■動作するためのパラメータの定義

部数、ページ数、両面、、、といった印刷パラメータや、Jam、用紙切れといった異常の発生の有無。
ターゲットモジュールの振る舞いに関する深い知見が必要

検証モデルの構成要素と作成の現状

構成要素	作成担当者	記述言語	Promelaへの変換方法
内部モデル	開発担当者	Model	自動
外部モデル			
タスク･イベントの管理	Specialist 開発担当者	Promela	手
他モジュールの振る舞い	Specialist 開発担当者	Promela	手
動作するためのパラメータの定義	開発担当者	Promela	手

ねらい

Specialist ⇒ 消去
Promela ⇒ Model
手 ⇒ 自動

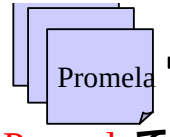
11. 外部モデル作成の専門性の排除の結果

THE DOCUMENT COMPANY FUJI XEROX

従来

外部モデル作成

Text Editor

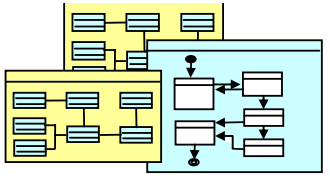


＋ Know How

Promelaで記述

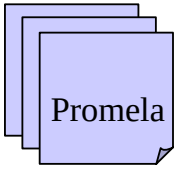
内部モデル作成

Modeling Tool

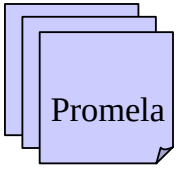


自動変換

検証モデル



＋



問題点

1.Promelaを知らないと駄目

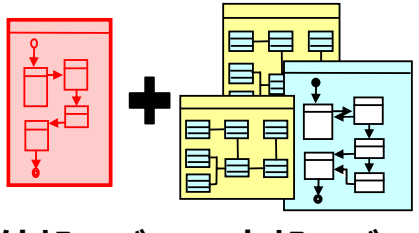
2.外部モデル作成KnowHow
タスク/イベントの管理･他モジュールの振る舞いの抽象化時にSPINの特質を考慮したタスク設計ができないと駄目

3.TextEditorで書くので見通しが悪い

改善後

検証モデル作成

Modeling Tool

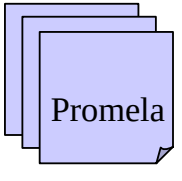


外部モデル　内部モデル

自動変換

Know How

検証モデル



解消後

1.Promelaを知らなくて良い

2.外部モデル作成KnowHow不要

3.Modeling Tool内で記述

12. やった事2

THE DOCUMENT COMPANY
FUJI XEROX

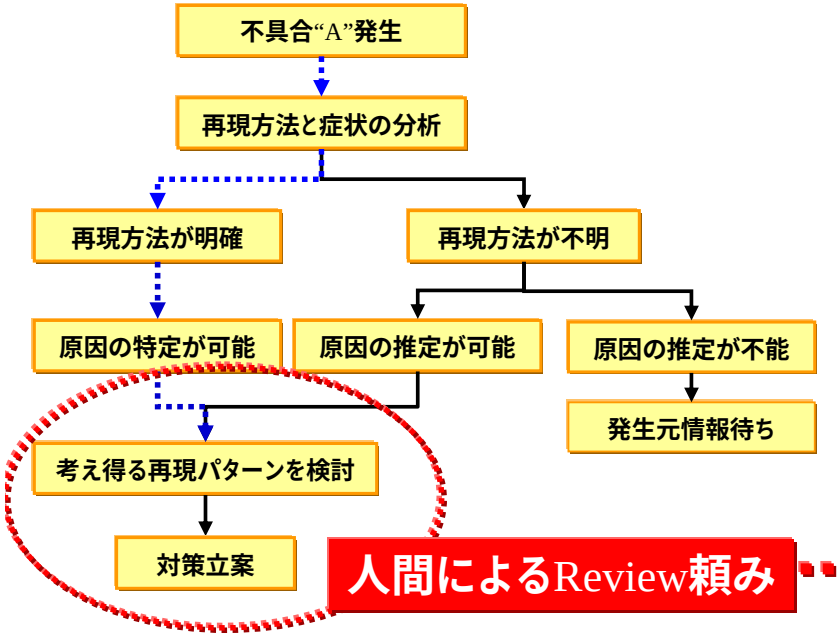
やった事　2　：　不具合“A”の対策検証

Model Checkといえども、使い方を誤っては逆効果
Model Check技術を使いながら適切に検証を進めるための方法を報告する

13. Model Check適用フロー

THE DOCUMENT COMPANY FUJI XEROX

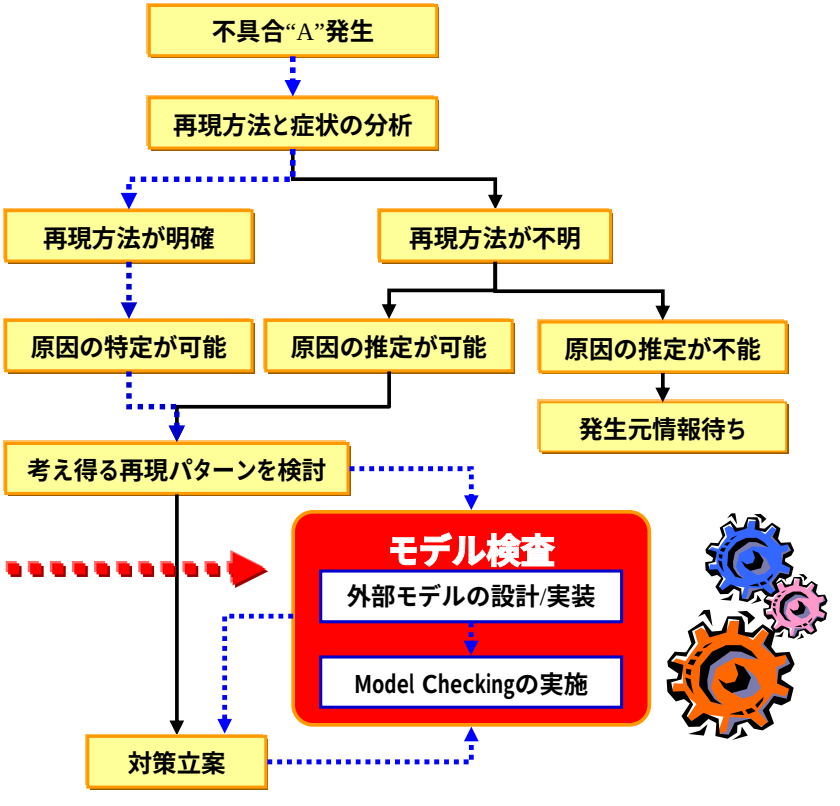
従来の不具合対応プロセス



開発フェーズ終盤、市場導入後で不具合が発見された場合、対策立案/再設計で、検討もれは御法度!!

限られたQCDで、2次不具合を出さない手法が必要。

Model Check導入時のプロセス

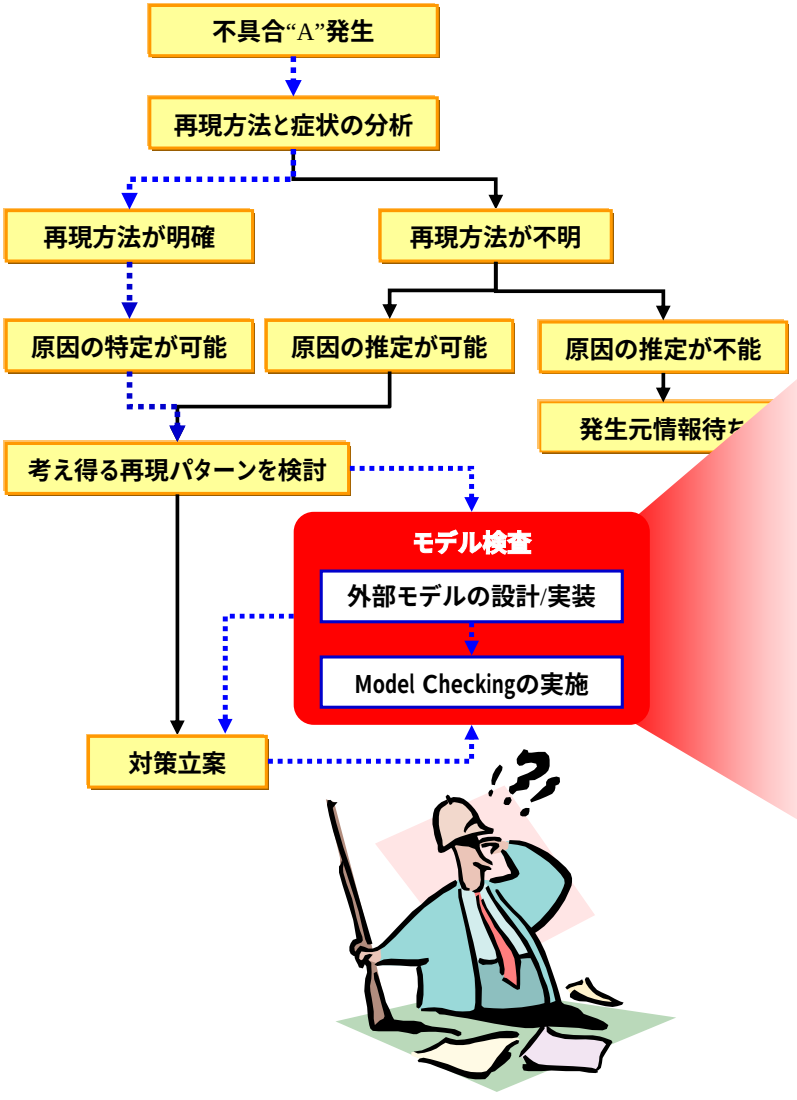


Model Checkで全空間網羅でテストすることにより、対策が十分であることを確認する。

14. Model Checkと外部モデルの信頼性

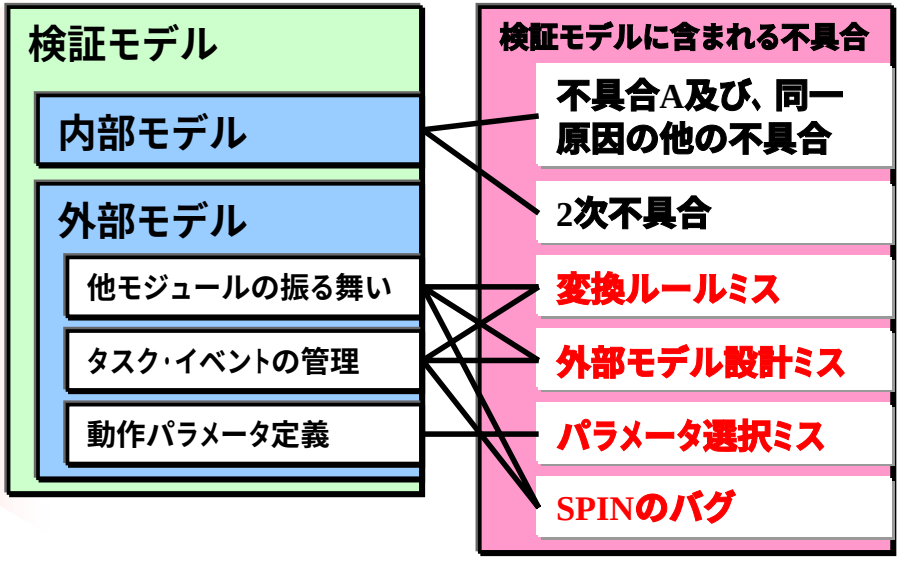
THE DOCUMENT COMPANY FUJI XEROX

だが しかし、、、Model Checkの導入による新たなリスク



新たなプロセスを導入するということは、
新たなリスクを抱えるということ。

Model Checkを実施するためには、
外部モデルの信頼性を確保しなければならない

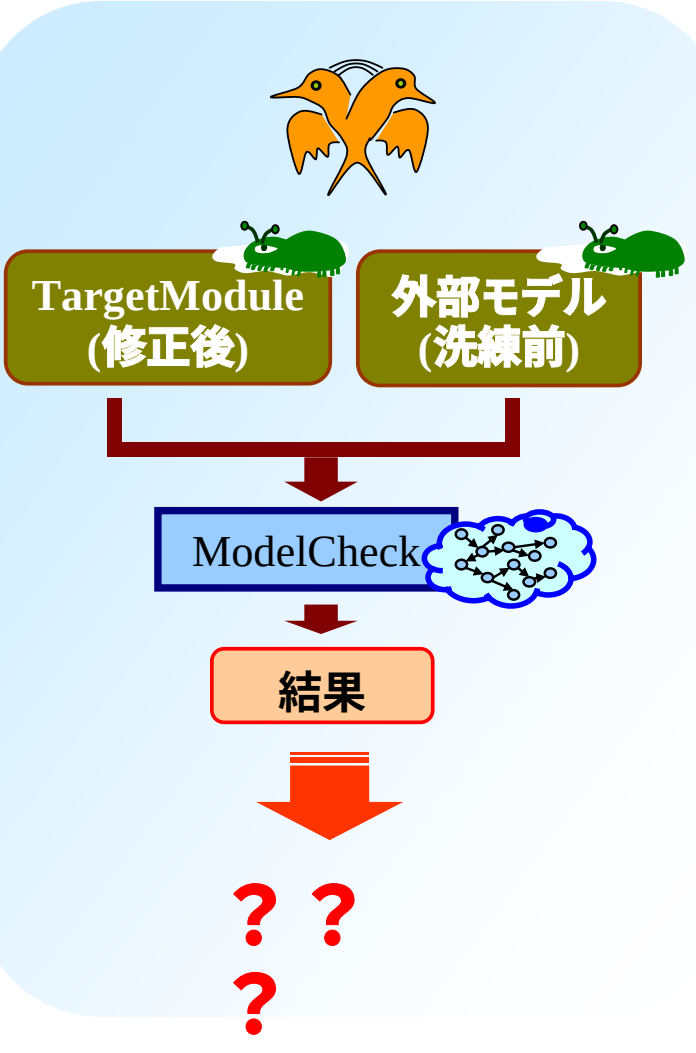


道具は常に使い方次第

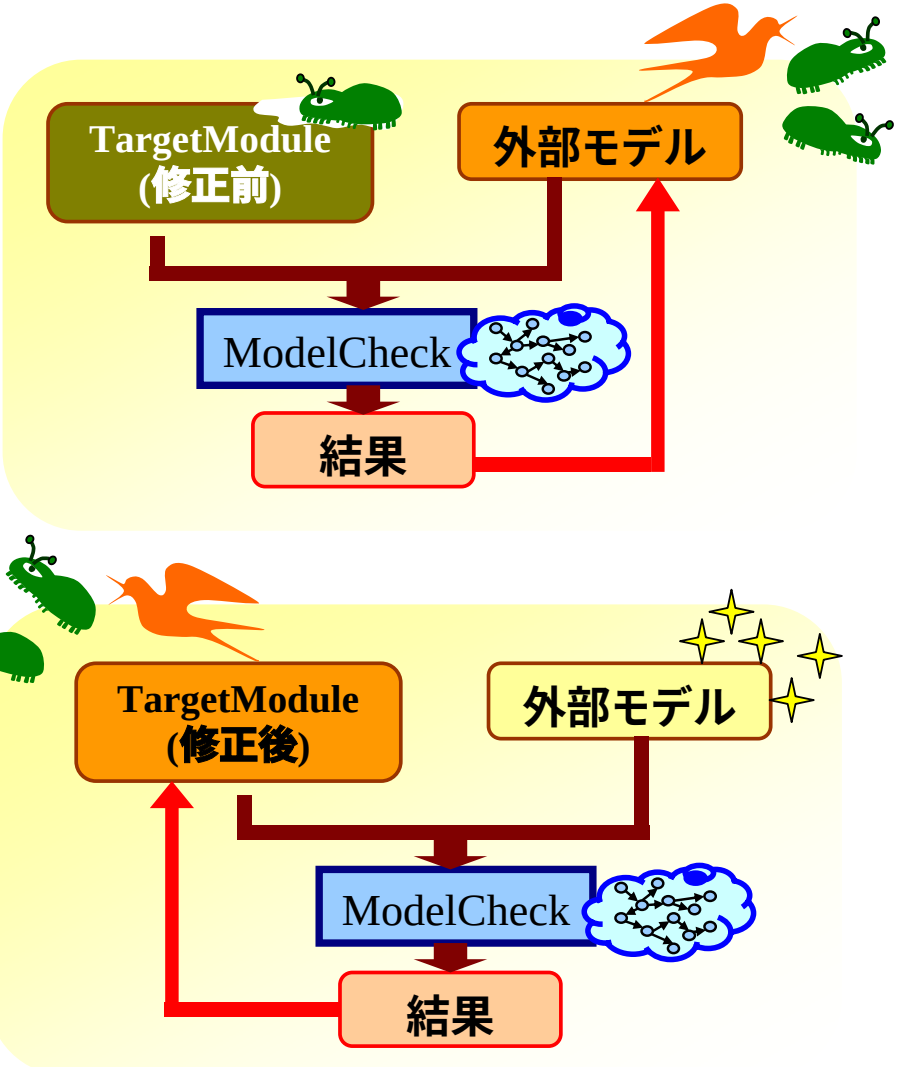
15. 外部モデルの信頼性確保

THE DOCUMENT COMPANY FUJI XEROX

何を検証している？



外部モデルの洗練



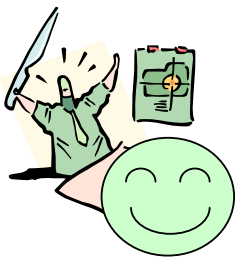
16. Model Checkの結果をどう解釈するか？

THE DOCUMENT COMPANY FUJI XEROX

不具合“A”の検証モデルでModel Checkを実施した場合の、検証結果のバリエーション

1. 想定した再現パターンだけが確認できる
2. 想定した再現パターンに加えて別な再現パターンが検出される
3. 想定した再現パターン以外のパターンだけが検出される
4. 不具合の検出ができない

外部モデルは正しい



外部モデルが小さい

外部モデルは間違っている

外部モデルは間違っている

実Target Moduleの状態空間

検証モデルの状態空間

1 は、要注意。偽ASSERTを仕込み、必要な空間を探査できているかどうかをチェックする必要有り

2 は、外部モデルがあっているのかどうかを確認する必要がある。別な方法が必要 (⇒次ページ)

3 4 は、ModelCheckの結果の通り解釈。外部モデルが間違っているので、ログを元に作り直し

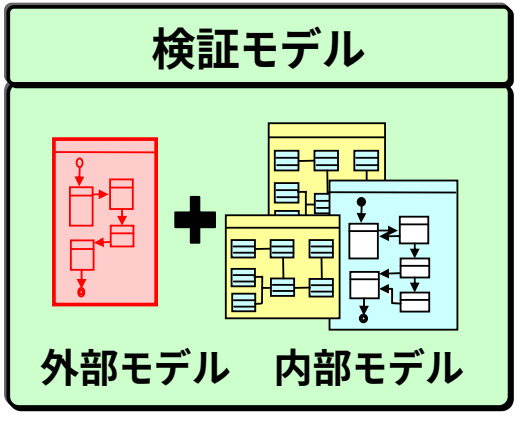
17. 他にも不具合パターンが検出される場合

THE DOCUMENT COMPANY FUJI XEROX

2 想定した再現パターンに加えて別な再現パターンが検出される

実行ログの取得

Modeling Tool上でログに従ってシミュレータでStep実行



ModelCheck	Step実行	解釈
○	○	新規の再現パターンとして認定
○	X	外部モデルの状態空間が過剰
X	○	検証手順上ありえない
X	X	検証手順上ありえない

18. Model Check 適用結果

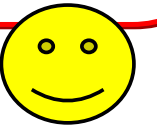
THE DOCUMENT COMPANY FUJI XEROX

外部モデルの信頼性確認結果

	再現パターン1	再現パターン2	再現パターン3
報告されている不具合	○	−	−
人間が原因から他の再現パターンを検討した結果	○	○	−
Model Checkによる検出結果	○	○	○
Modeling Tool 上での、Step実行による確認結果	○	○	○

MCで新発見!

外部モデル信頼性あり



Model Checkに要した各工程とその工数

作業内容	担当	期間
Modeling Toolでの外部モデル作成	1名	3週間程度（2～3時間/日）
外部モデルデバッグ	1名	1週間程度（2～3時間/日）
不具合Aの再現テスト	1名	3日（2～3時間/日）
不具合A対策版での回帰テスト	1名	半日

スケジュールインパクトなくできた

19. まとめ

THE DOCUMENT COMPANY FUJI XEROX

ModelCheckingからの専門性の排除

ターゲットモジュールのドメイン知識さえあれば
ModelCheckingが使える仕組みを作った。

- 「なにをやるか」と「どうやってやるか」を分離
- 開発担当者は「なにをやるか」に専念できる

簡単！

外部モデルの信頼性確保

外部モデルの信頼性を確保する施策を導入した。

- 想定した再現パターンとModelCheckingの結果を比較して外部モデルが正しいかどうかを確認する。

安心！

不具合Aの検証に適用

- 対策の効果を確認。二次不具合なし。
- 対策導入時の不安(少し)解消！！

ModelChecking
は楽に使える！

20. 今後の方向性

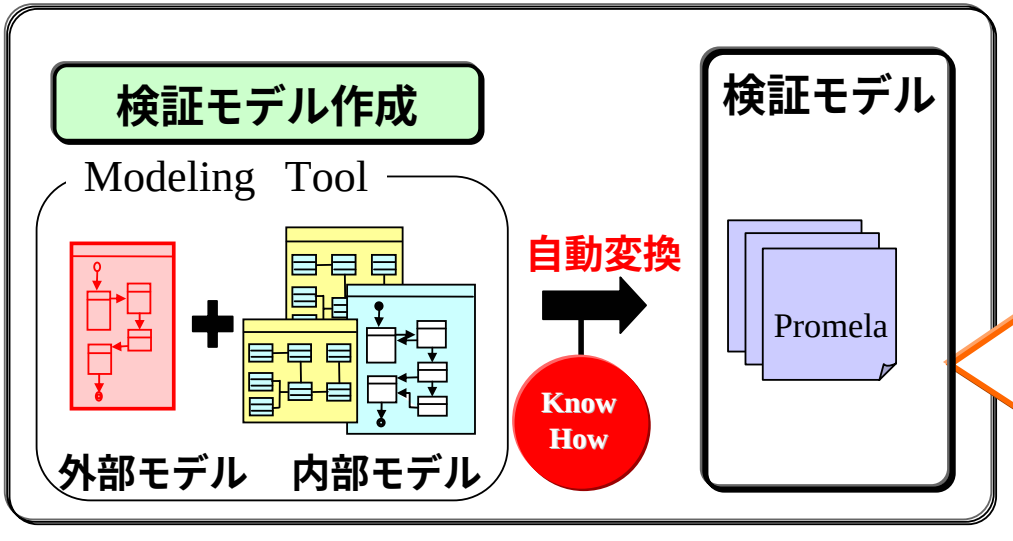
THE DOCUMENT COMPANY FUJI XEROX

Early Bird As The 2nd Eye.

第2の目

人間はどうやったって、ミスをする。
ローマの時代から変わらない本質。
それを素直に認めて機械の目をもう一つ持っていても良いのでは？
積極的に製品開発へ応用を検討する。

Model Checkingの普及



まだ、開発環境としては、未熟。
SPIN、Cygwin等の束縛がある。
完全な統合環境が欲しい。

今後は統合環境の開発を検討する方向。
それが、日本のSW開発者の為になるのであれば、我々は、その開発環境を提供することを惜しんだりはしない…と思う

ご清聴ありがとうございました。