企業におけるフォーマルメソッドの実践

～始めよう！広げよう！モデル検査！～

メルコ・パワー・システムズ株式会社
技術統括部　ビジネスチーム２
早水公二


モデル検査を始めるには


初めての社内適用


どのように広めたか？


使うと何が嬉しい？


始めてみませんか？


事例紹介

1 モデル検査を始めるには

新しい手法は敬遠される？

経験豊かな技術者(職人)

従来手法で成功した人

モデル検査は少し時間が必要

(導入時の勉強／適用時のモデル化作業)

S/W開発現場は忙しい！

大企業で多額の研究費と人手をかける方法もあるが．．．

まずは“自分”が率先して！

勉強してみましょう

1 モデル検査を始めるには→ まずは自分が率先して

モデル検査に関する書籍

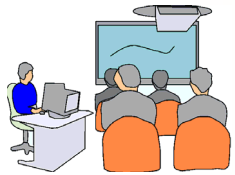
「4日で学ぶモデル検査(初級編)」
産業技術総合研究所システム検証研究センター 著

「SPINモデル検査ー検証モデリング技法」
中島 震 著

ご参考までに

「モデル検査器SMVガイドブック」
モデル検査によるソフトウェアテストの実践研究会 著

不定期のセミナー



- ☑ 日本科学技術連盟主催
- ☑ 情報処理推進機構(IPA) SEC主催
- ☑ 産業技術総合研究所システム検証研究センター主催
- ☑ 日本ソフトウェア科学会主催

1 モデル検査を始めるには

結果を出してみないと効果が見えにくい

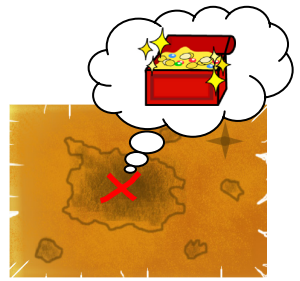
モデル検査は？

モデルと検査式との“不整合を発見”する手法

⇩

ソフトウェア開発に適用すると？

成果は“不具合を発見”すること(広義には品質の向上)

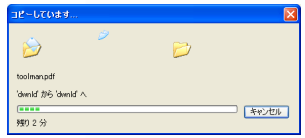


やってみないと分らない

まずは“実践”してみる！

習うより慣れる

モデル検査器のダウンロードができるページ



- NuSMV
  http://nusmv.irst.itc.it/
- SPIN
  http://spinroot.com/spin/whatispin.html
- UPPAAL
  http://www.uppaal.com/

1 モデル検査を始めるには

前もって効果を数値化しにくい

発症前に発見した不具合の
価値は測れない

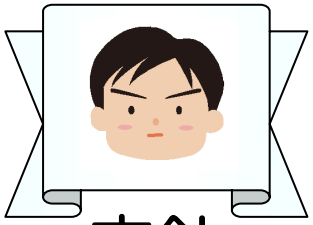
- もし出荷後に発症してたら？
- もし緊急事態で発症してたら？
- もし従来手法で見つけるとしたら？

多くのシステムに適用して
多くの実績を挙げる

粘り強く“継続”する！

実績を出せば
効果は認められる

2 初めての社内適用　～体験談～



方針

まずは自分で実践して効果を体感して
社内にアピールしよう！！

理屈より結果を出すのが一番！

- 初適用の対象は
  社内イントラシステム
  (勤怠記録システム)
  ▶ 全社員が毎日利用
- 検査対象は
  稀に(1回／月～)発生する
  ＤＢ登録時の排他ミス
  ▶ モデル検査の得意な分野
- 事前報告
  モデル検査の適用を
  社内で事前に報告
  ▶ 注目を集める

結果は？

不具合解析に成功

PR効果“大”

3　どのように広めたか？　～体験談～

- ☑ 不具合で困っている プロジェクトを探す ▶ 適用を打診

他部署の週報も チェック

先輩、後輩、同僚 にインタビュー

- ☑ 常に適用対象を募集
  休憩時間、昼食時、喫煙室、宴会．．．

- ☑ 適用結果は報告書に まとめた ▶ 社内で回覧 報告会も開催

「モデル検査の社内適用 成果報告書」

3　どのように広めたか？　～体験談～

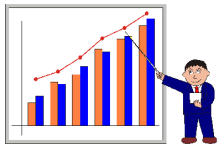
- ☑ 技術説明会も開催

- ☑ その他

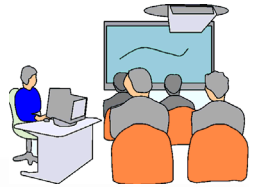
論文投稿
シンポジウム等での発表．．．

結果？

適用依頼が増加



手法を教えて欲しい！　→ モデル検査セミナー開講



コース	概要
基礎コース	基本的な理論と具体的な検査技術，モデル検査器の操作方法を教育する
応用コース	フローチャートで記述されたシステムの検査技術を教育する
実践コース	受講者による実システムへの適用を支援しながら，検査技術者を養成する

知識と技術を教える＋「人材」を育成する

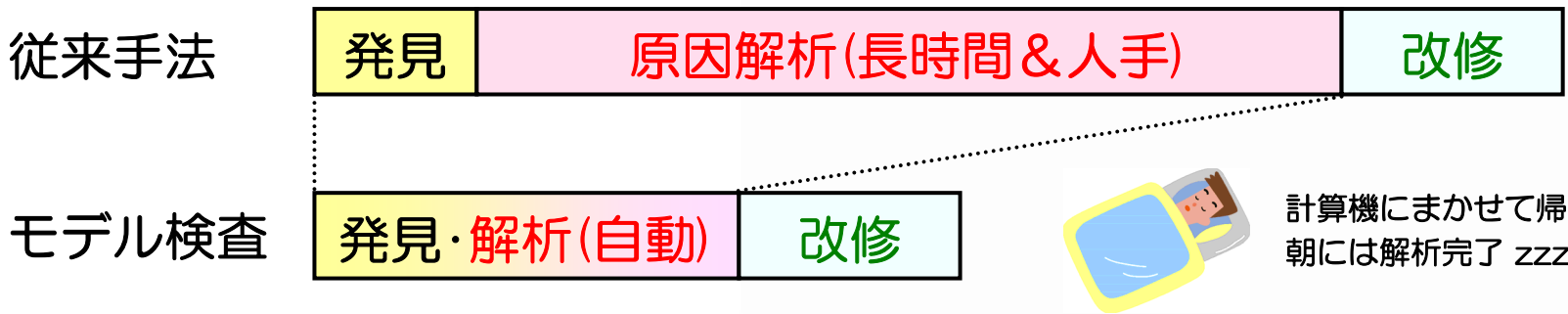
4 使うと何が嬉しい？ ～実感～

＜直接効果＞

想定外の不具合が見つかる ▶ 全数検査の魅力

Javaのスレッド切替のタイミングに起因する不具合
ユーザのダブルクリックのスピードに起因する不具合
設定値の組合せによって発生する不具合
割り込み処理のタイミング、順序によって発生する不具合
重故障発生時のあるタイミングで1回だけ実行される処理の不具合
ファイル定義書と実行モジュールとの不整合
…

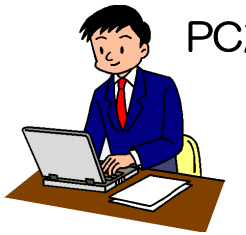
不具合に至る経路もわかる ▶ 発見と解析が同時



4 使うと何が嬉しい？ ～実感～

＜直接効果＞

- 不具合解析に使うと効率的 ▶ B/C ＝ 大
  （Benefit/Cost）
- 回帰試験はリターンキーだけ ▶ 何度でも全数検査
- 実機は不要 ▶ 組込みでは効果的

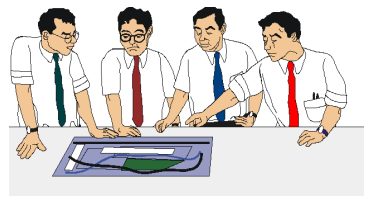


- 反例ファイルの共有 ▶ 不具合情報の共有

反例ファイルとモデルがあれば、いつでも不具合現象を再現できる
プロジェクトメンバで情報共有すれば横展開・再発防止ができる

▶ 障害管理にも利用可

4 使うと何が嬉しい？ ～実感～



＜副次的効果＞

- モデル化のため仕様書を熟読 ▶ 設計レビュー
  - モデル作成中に仕様書の不備／矛盾が見つかることもある
  - 仕様書の重要性を再認識する
- 仕様を厳密に規定 ▶ 設計能力の向上
  仕様記述力の向上

＜自動販売機の仕様＞

□お金を入れてAのボタンを押すとAの商品が出る。
→Aのボタンを押しながらお金を入れるとどうなるの？

□お金を入れて返却レバーを下すと返金される。
→停電時の動作は？　正しく返金されるの？

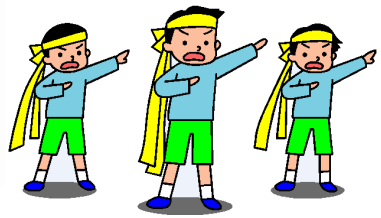
4 使うと何が嬉しい？ ～MPSの場合～

メルコ・パワー・システムズの社内適用事例

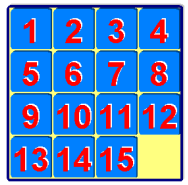
システム名	適用結果
勤怠管理	入力操作のタイミングに起因する不具合の原因を究明
広域情報監視	非同期プロセスの割込み処理に起因する不具合の原因を究明
電力設備保全(Ver.1)	仕様の不備1件と不具合３件を発見
輸送運行情報監視	条件分岐処理の不具合２件を発見
画像認識（事象検出）	通信異常時の初回処理だけに発生する不具合の原因を究明
広域システム共通Ｌｉｂ	運用で用いるファイル定義書の不備１件を発見
電力設備監視	波形解析処理のアルゴリズムに起因する不具合の原因を究明
電力設備保全(Ver.2)	不具合の原因を究明し，さらに別の不具合２件を発見
＜その他＞基幹系システム６件　組み込み系システム１件　現在適用中１件	

不具合は、従来の動作試験では発見あるいは原因を究明できなかったもの

続ければ応援してくれる！



5　始めてみませんか

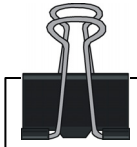


❀ まずは簡単なモデルから

→ 身近なもの(自動販売機、パズル、クイズ．．．)をモデル化して検査する
→ 難しさを区別（業務？　モデル検査？）

❀ 慣れたら業務に適用

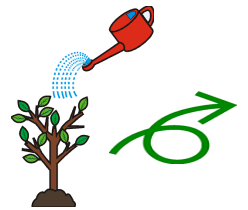
→ どの工程？　（ 要求仕様　機能仕様　詳細仕様　ソースコード ）

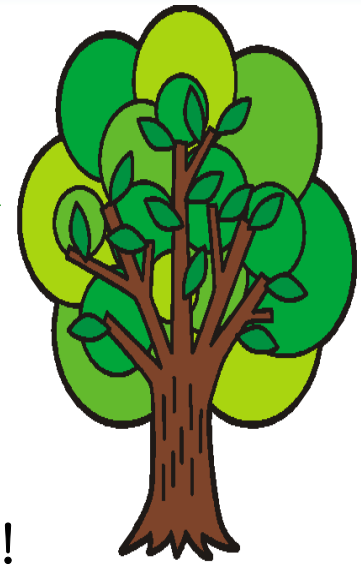


モデル検査は上流工程向きと言われるが．．．

自然言語(日本語、英語)、概略フロー等で記述された
仕様は曖昧性が高く、モデル化するのが難しい！
一意に読み取れるソースコードに適用する方が意外と簡単

5 始めてみませんか





❀ 小さなソースコードからコツコツと

→ １００～２００行程度で練習

→ C言語がお勧め （Java、C++等は少し工夫が必要）

→ ブロック、関数、小機能．．．だんだん大きくする

→ いきなり大きな対象に挑戦すると途中で挫折することも！

❀ 不具合が見つかったら？ ▶ モデルと反例は保存しておく

→ ある製作者：「あの不具合ってどんな現象だったっけ？」

→ モデルと反例があれば、いつでも現象を再現できる

（直接効果；反例ファイルの共有）

5 始めてみませんか

❀ 検査式の習得が **カギ**

→ プログラム経験者：モデルの作成は比較的簡単
→ 日本語から検査式(CTL式、LTL式等)への変換が難しい

検査式は１か所間違うと全く異なる意味になる！！

AG(EF(初期状態)) ：いかなる状態からでも初期状態に戻れる
EF(初期状態) ：初期状態から初期状態に戻れる(自明：必ずTrueになる)

▶ 検査式は”きっちり”勉強する

❀ 成果を出せたら？ ▶ 大々的にアピール！

▶ 局所的な適用から
全社導入へ