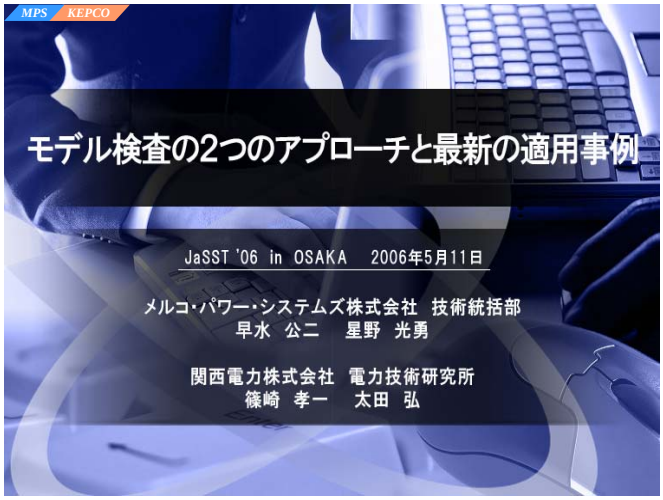


MPS KEPCO

発表の内容

- モデル検査の社内適用→事例分析
- 適用の際の２つのアプローチ
- 事例報告
- 今後の取り組み

MPS KEPCO

モデル検査の社内適用

＜モデル検査技術者＞
社内に５名のモデル検査技術者

＜体制＞
モデル検査の研究・社内適用に２名の専任者

専任技術者Ａ（モデル検査歴　４年）
専任技術者Ｂ（モデル検査歴　１．５年）

MPS KEPCO

モデル検査の社内適用

自社開発した１５件のシステムにモデル検査を適用

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	

MPS KEPCO

適用事例の分析

モデル検査を適用した１５件の事例を
適用時の形態に着目して分類

規則性を発見 ⇨ 定型的な適用手法

適用時の形態 = 設計者からの依頼形態

◆不具合はないが、モデル検査の網羅的な
デバッグで品質を向上させたい（品質向上）

◆既に発生してしまった原因不明の不具合を
モデル検査で解析してほしい（不具合解析）

MPS KEPCO

適用事例の分析

分類結果は（代表的な好事例８件）

システム名	適用形態	
	品質向上	不具合解析
勤怠管理		○
広域情報監視		○
電力設備保全(Ver.1)	○	
輸送運行情報監視	○	
画像認識（事象検出）		○
広域システム共通Lib	○	
電力設備監視		○
電力設備保全(Ver.2)		○

MPS KEPCO

適用事例の分析

さらに形態毎に特徴を抽出すると

適用形態	システム名	適用時期	検査対象の範囲
品質向上	電力設備保全(Ver.1)	設計時から運用開始前	極力広い範囲
	電力設備保全(Ver.1)		
	輸送運行情報監視		
	広域システム共通Lib		
不具合解析	勤怠管理	本格運用前	特定の狭い範囲
	広域情報監視		
	画像認識(事象検出)		
	電力設備監視		

MPS KEPCO

２つのアプローチ

適用の際の２つのアプローチを提案

◆品質向上のアプローチ

◆不具合解析のアプローチ

MPS KEPCO

モデル検査の適用手法の提案

適用にあたっての２つのアプローチを提案

◆品質向上のアプローチ

開発初期から運用開始までの全般にわたって、広い範囲を検査し１つでも多くの不具合を発見することによって品質向上を目指す

→ 従来手法と並行してモデル検査を実施

◆不具合解析のアプローチ（デバッグへの応用）

特定の狭い範囲に的を絞って、本格運用前など緊急時に発生した原因不明の不具合を早期に解析する

→ モデル検査に重点を置いて実施

２つのアプローチを使い分ける

→適用時期・不具合の有無等によって
モデル検査と従来手法とを使い分ける

MPS KEPCO

モデル検査の効率的な導入形態

設計段階 ⇩

仕様書やフローチャートがあればモデル検査可能

モデル検査を単独で使う（品質向上）

S／W製作中 ⇩ S／W試験中

複雑でない不具合は従来手法でも発見できる

微妙なタイミング・複雑な条件で発生する不具合はモデル検査で発見できる

システムの中枢部分だけモデル検査を行う

従来手法との併用（品質向上）

本格運用前

原因不明の不具合発生！　原因がわからない！
調査する時間がない！早く解決したい！

モデル検査を単独で使う（不具合解析）

MPS KEPCO

モデル検査を導入しやすいシステム

モデル検査は微妙なタイミングや、複雑な条件が重複して発生する不具合を発見するのに有利

○タイミングが重要な通信システム
○不確定要素（人間の操作等）を含むシステム（HMI系など）
○割り込み処理の多い組み込み系システム
○起動タイミングが複雑なマルチスレッドで動作するアプリケーション
○アルゴリズムが非常に複雑な、基幹系システムの中枢機能

通信システムを除く上記全てのシステムで適用実績有り

MPS KEPCO

モデル検査を導入しにくいシステム

○アルゴリズムが簡単でも大量のデータを扱うシステム
→ 状態爆発
○膨大なページ数の仕様書、大きなソフトウェア
→ モデル化の作業量大
○曖昧な日本語で記述された仕様書
→ モデル化できない
○再帰処理のあるシステム
→ 無限系は理論的に検査できない
○乗除演算（×÷）の多いシステム
→ BDD（Binary Decision Diagram）に不向き
○人間の思考回路に適さないシステム（クイズ、ゲーム等）
→ 同上

今後導入できそうな分野

モデル検査から出力される反例を活用できる分野
反例＝ある状態にたどり着くまでの経路

- ○カーナビの経路探索
- ○電力系統制御での迂回路探索
- ○列車のダイヤ改正
- ○Webシステムの画面遷移の確認

網羅的な検査法を活用できる分野

- ○リアルタイム処理のDeadLine判定（エンジン制御、鉄鋼プラント等）
- ○法律・規則の「抜け道」の有無

事例報告（不具合解析のアプローチ）

物質検出システム（※システム名、物質名は仮称）

金、銀、銅、アルミニウム、水銀…を検出し表示するシステム

設計者から「原因不明の不具合をモデル検査で解析してほしい」

事例報告

原因不明の不具合

A工場に設置したシステムでは、「ニッケル有り」は表示する
必要がないのに、一度だけ表示してしまった。
（システムを設置する工場によって表示する物質の種類／数が異なる）

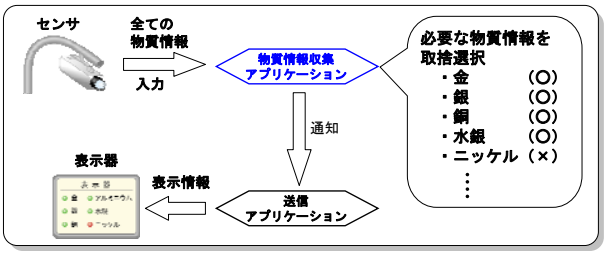
なぜ原因がわからなかったか？

■発生状況が不明確
- ハードウェアの試験中に表示されたらしい
- ハードウェア担当者はそれが不具合だとは知らなかった
- 不具合発生時にソフトウェア設計者が不在であった
  →再現試験ができない
- 設計者：表示器の故障ではないのか？

設計者とプログラマーが２日間コードレビューをしたが原因は不明…

事例報告

物質情報収集～送信までの流れ



モデル検査の対象 ＝ 物質情報収集アプリケーション

事例報告

	作業時間
1 物質情報収集アプリケーションの仕様理解	１６時間
2 フローチャートを作成	３時間
3 モデルの作成（モデル検査支援ツールにより自動生成）	作業なし
4 検査項目（CTL式）の作成	数分
5 モデル検査器の実行	数分
6 反例の解析	０．５時間
	２０時間

事例報告

検査対象とした物質情報収集アプリケーションは

言語　：C言語
行数　：５０３２行（コメント行含む）

検査に使用した計算機のスペック

CPU　：３．０[GHz]
メモリ　：２．０[GB]

検査項目（CTL式）

```
!EF (Nickel = 1 & Set_Find = 1)
```

EF：Exist Furture
存在　将来

式の意味は？
「ニッケル有り(Nickel = 1)で、かつ、検出通知有り(Set_Find = 1)の
状態は将来にわたって存在しないはずである」

MPS KEPCO

事例報告

モデルと検査項目

```
MODULE main
  VAR
    pjdg_jdg  : boolean;
    chgflg    : boolean;
    Nickel    : boolean;
    Set_Find  : boolean;
      ⋮
  ASSIGN
    init(pjdg_o_jdg) := 0;
    next(pjdg_o_jdg) := case
        PC = 17 : pjdg_jdg;
        PC = 20 : pjdg_jdg;
        1 : pjdg_o_jdg;
    esac;
      ⋮
SPEC  !EF(Nickel = 1 & Set_Find = 1)
```

Cソースコードから作ったモデル

← 検査項目

MPS KEPCO

事例報告

検査結果 ＝ False

ニッケル有りで検出通知有りの状態が存在する！

反例（不具合状態に至るまでのログ）

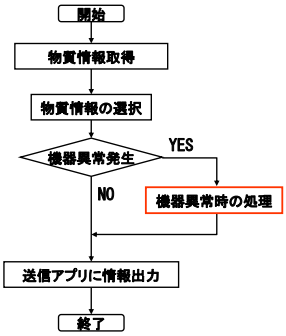
変数	0		1	2	3	4	5	6	7	8	9	10
Nickel	0		0	0	0	0	0	0	0	1	1	1
PC	1		6	9	10	11	14	15	18	16	12	13
Set_Find	0		0	0	0	0	0	0	0	0	0	1
Jdgdata	0		1	1	1	1	1	1	1	1	1	1
pJdg_jdg	0		1	1	1	1	0	0	0	0	0	0
⋮												

PC（プログラムカウンタ）：フローチャート中の位置を特定

MPS KEPCO

事例報告

不具合の箇所



MPS KEPCO

事例報告

```
機器異常時の処理( ){
    ⋮
  物質情報の再取得( )
  物質情報の再選択( )
    ⋮
}
```

→

```
物質情報の再選択( ){
    ⋮
 /* 物質情報のマスク処理 */
    物質情報 = センサ情報   PC = １６
 /* 物質情報 = センサ情報 & 0x052F */
    ⋮
}
```

コメントアウトのまま！（これが不具合原因）

MPS KEPCO

事例報告

不具合の理由　機器異常時の処理

A工場に設置するシステムはB工場のシステムを改良したもの
（A工場とB工場では設計／製作担当者が異なる）

A工場とB工場のシステムには特殊仕様が存在した
→機器異常発生時に１回だけ物質情報を再取得し、さらに物質情報の再選択を行う

B工場では全ての物資が表示対象であった
→再選択処理で物質情報（ビット情報）の
　マスク処理は不要（コメントアウトした）

物質情報　0000 0101 1010 1111
ニッケル

A工場ではニッケルは表示対象外である
→A工場担当者は、ニッケルのビットをマスクする必要があるのに忘れていた…

MPS KEPCO

事例報告

不具合が発生した状況

①ハードウェア（H／W）の試験で機器異常時の動作試験を行った
②試験用のセンサ模擬データがセットされたままになっていた
　→本来は、H／Wの試験では模擬データは不要
③模擬データの一部に「ニッケル」のデータが含まれていた
④機器異常時に特殊仕様の処理（物質情報の再取得）が動作した
⑤再取得した１回分の模擬データに偶然「ニッケル」が含まれていた
⑥１回だけ「ニッケル有り」を表示してしまった

モデル検査だから原因を究明できた

約２０時間で５０００行の中から不具合箇所を特定

（仕様理解を除けば数時間）

今後の取り組み

◆事例の積み重ねツールの開発を継続

◆事例の分析

→ モデル検査の適用基準策定 → 導入指針

◆事例・ツールの紹介

→ モデル検査のＰＲ・普及促進

ご清聴ありがとうございました