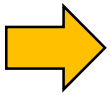
IPA Software Engineering Center
Information-technology Promotion Agency, Japan

実稼働システムを対象とした形式手法適用実験について

2012年 11月 12日

情報処理推進機構 (IPA)
ソフトウェア・エンジニアリング・センター (SEC)
連携委員　向山　輝

内容



- **形式手法適用実験の背景と目的**
- **実施した実験の概要**
- **実験結果**
- **まとめ**

背景・環境

SEC
Software Engineering
for Mo・No・Zu・Ku・Ri

『形式手法』に関する認識と環境

- 高信頼なソフトウェア開発に有効な手法（といわれている）
- 国際規格で利用を推奨（IEC15408，IEC61508，ISO26262，EN50128など）
- 欧州で形式手法の産業利用に積極投資
  - EUの政策の一環として、形式手法の普及を財政支援（*）
    鉄道、自動車、航空宇宙、原子力などで形式手法の利用が進む。

（* 第7次研究枠組み計画（FP7），2007-2013年）

- 経産省「情報システムの信頼性向上に関するガイドライン」で形式手法の活用を推奨（2006年）
- ソフトウェアの不具合によるシステムトラブルの社会問題化

形式手法に対する期待・関心は高まっている

これまでの取組みと普及状況

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

IPA SECの取組み

- 2007年：「高信頼性システム技術調査検討会」発足
  - 海外、国内の形式手法の導入事例調査
- 2010年3月：「高信頼性システム開発技術の動向」を公開
- 2010年7月：「形式手法適用調査」報告書を公開
- 2011年3月：「形式手法導入のための教材」を開発
  - 広島、北海道などで研修を実施

国内の形式手法普及状況

- **広く開発現場に普及しているとはいいがたい**
- **とくにエンタプライズ系では、適用事例が少ない**
  （IPA SEC「形式手法適用調査」報告書の調査事例103件中情報システムは3件）

形式手法適用実験の目的

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

形式手法が適切に活用されるようになるためには・・・

解決が必要な課題

形式手法導入の検討をするために必要な情報が不足

- 形式手法の効果、影響が明らかでない
- 形式手法適用の決断ができない　→　適用例が増えないため情報が増えない

以下を具体的に明らかにすることを目的として、
実システムを対象とした形式手法の適用実験を実施

- 形式手法を適用した場合にどのような**効果**があるか
- 形式手法の適用にはどの程度の作業**工数**が必要か

内容

- 形式手法適用実験の背景と目的
- **実施した実験の概要**
- 実験結果
- まとめ

実験対象とした形式手法の使い方

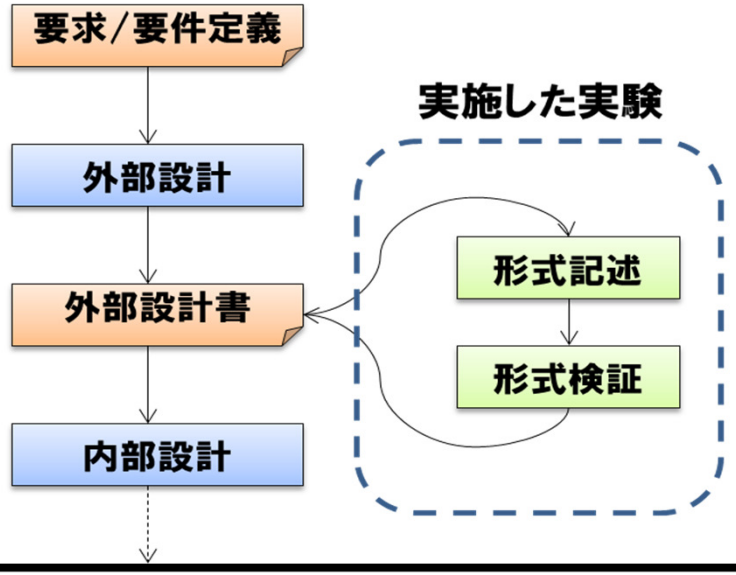
- **形式手法の使い方は大きく2つ**
  - **構築(Construction):**

    形式仕様言語よるデザイン記述を得ることを目的とする。形式検証で記述を確認しながら構築していく。

  - **検査(Inspection):**

    与えられた仕様書の内容を確認することを目的とする。仕様書を形式記述に変換し、検証する。

今回の実験は、外部設計書の『検査』に形式手法を使う方法で実施



実験対象と実験方法

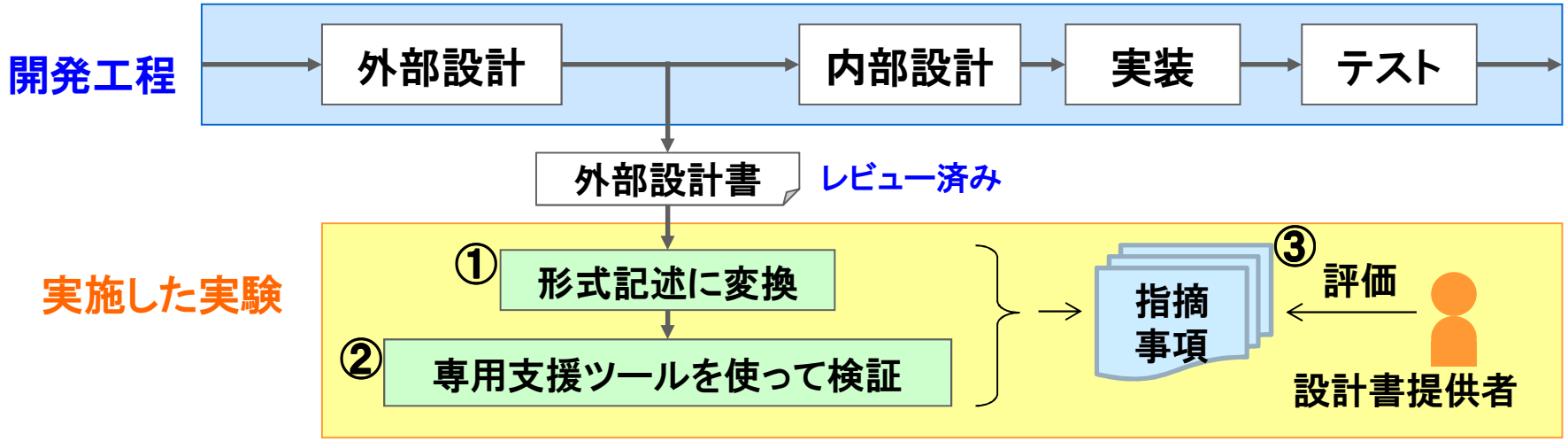
SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

■ 実験対象とした設計書

- 東京証券取引所で開発され、運用されている情報システムの設計書を対象
- 外部設計終了（レビュー完了）時点のもの

■ 実験の方法

① 設計書を形式仕様言語による記述（形式記述）に変換
② 専用の検査支援ツールを使って検証
③ この作業で見つかった指摘事項をリストアップし、設計書提供者が指摘の妥当性を評価



形式手法による設計書の検査

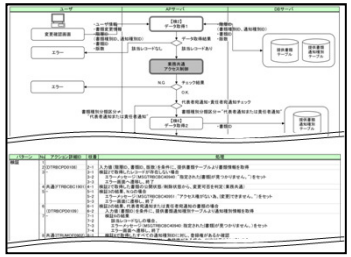
SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

■ 形式手法を使う手順

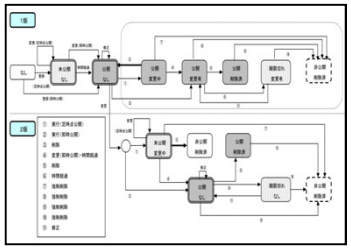
- DSF(※)が公開した『形式手法活用ガイド』に示される手順で実験を実施

複数の設計書に書かれた仕様を形式仕様言語で記述し、整合性などを検査

例



画面アクション明細



状態遷移図

形式仕様言語で記述

```
: if

状態「なし」の場合
  /* なし */    :: atomic{
    (CmnDualPst == Pini) && (CmnDualDst == Dntg) ->
    if
      /* なし -> 未公開-なし(登録(定時)) */
      :: event == Ergs -> CmnDualPst = Punl
      /* なし -> 公開-なし(登録(即時)) */
      :: event == Ergn -> CmnDualPst = Plst
    fi }

状態「未公開-なし」の場合
  /* 未公開-なし */    :: atomic{
    (CmnDualPst == Punl) && (CmnDualDst == Dntg) ->
    if
      /* 未公開-なし -> 公開-なし(変更(即時公開)) */
      :: event == Echn -> CmnDualPst = Plst
      /* 未公開-なし -> 非公開-削除済(削除||強制削除) */
      :: (event == Edlt || event == Efdl) -> CmnDualPst = Pnds;
CmnDualDst = Ddlt
      /* 未公開-なし -> 変更(定時点公開) */          :: event ==
Echa -> skip
    fi }
```

形式記述

•設計書記述の不足、曖昧性
•設計書間の不整合
を検査

画面アクションの内容で、
状態遷移図で書かれた通りに
データが変化するか?

検査支援ツールで検証

(※ ディペンダブル・ソフトウェア・フォーラム: 設計品質の向上手段として形式手法に着目し、開発現場への普及展開を目的として、民間企業6社1機関が共同で活動。)

（参考） DSF：ディペンダブル・ソフトウェア・フォーラム

SEC
Software Engineering for Mo·No·Zu·Ku·Ri

- **エンタプライズ系ソフトウェアの信頼性を向上する技術の研究開発を行う産学連携フォーラム** http://www.nttdata.co.jp/dsf/index.html
  - メンバ：NTTデータ，富士通，NEC，日立，東芝，SCSK，NII
  - 活動期間：2009年12月 ～ 2012年6月

- **産業界の実績、ツールの実用性などの観点で3手法を選び、使い方のガイドを作成**
  - VDM++，Event-B，SPIN
  - 「形式手法活用ガイド」・・・ 手引き、手順、イディオム集

本実験は、このガイドを参照して実施

- **活動状況**
  - ‘09年．図書館システム（開発教育用）を用いた記述実験
  - ’10-11年．形式手法活用ガイド【ドラフト版】【リリース版】を公開
  - ‘11年．実システム（東証）の設計を対象とするIPA実証実験に参加
  - ’12年．形式手法活用ガイド【改訂版】を公開。IPAに委譲し、活動終了。

実験実施体制

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

- IPA SEC 「形式手法適用実証ワーキング・グループ（WG）」の活動として実施
- 実験期間： 2011年8月 ～ 2012年3月
- 参加メンバ（WG委員）:

立場	メンバ	実験での役割
ベンダ	株式会社NTTデータ	形式手法の適用
	富士通株式会社	
	日本電気株式会社	
	株式会社日立製作所	
	株式会社東芝	
	SCSK株式会社	
ユーザ	株式会社東京証券取引所	設計書の提供 指摘事項の評価
	住友電気工業株式会社	
学識経験者	九州大学	実験結果の評価
	国立情報学研究所	
	名古屋大学	

実験実施体制(続き)

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

使用した形式手法

- Event-B、SPIN、VDM++　の3種類。
  - 知名度が高く利用実績も多い
  - 解説書、支援ツールが手に入りやすい

実験チーム

- 形式手法の種別(適用法)に対応して、5つの実験チームを編成。
- 各実験チームが、独立に実験を実施。

チーム	実験対象設計規模(ページ数)		実施体制(人数)	形式記述(行数)
	形式記述作成対象ページ数	参照を含む総ページ数		
Event-B (1)	110	707	1	1.0K
Event-B (2)	106	287	3	0.4K
Event-B (3)	49	381	1	0.4K
SPIN	109	429	2	0.7K
VDM++	300	700	5	10.8K

（参考）実験者のスキル

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

チーム名	実施体制（人数）	作業者	役割*	スキル（経験）			
				業務AP開発	類似AP開発	形式手法	採用手法
Event-B 1	1	A	形式記述者 形式検証者	1年	0年	3年	2.5年
Event-B 2	3	A	形式記述者	15年	0年	3年	3年
		B	形式記述者	0年	0年	1年	1年
		C	形式検証者	5年	0年	7年	1年
Event-B 3	1	A	形式記述者	13年	5年	1.5年	1.5年
SPIN	2	A	形式記述者	0年	0年	6年	6年
		B	形式記述者	2年	0年	5年	0年
VDM	5	A	形式記述者 形式検証者	2年	0年	5年	5年
		B	形式記述者 形式検証者	2年	0年	5年	1年
		C	形式記述者 形式検証者	6年	6年	20年	2年
		D	形式記述者 形式検証者	0年	0年	3年	2年
		E	形式記述者 形式検証者	37年	25年	17年	12年

内容

- 形式手法適用実験の背景と目的
- 実施した実験の概要
- **実験結果**
- まとめ

指摘事項に対する設計書提供者の評価

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

実験で検出された指摘事項55件に対する設計書提供者の評価

設計書提供者による評価	件数
設計書の修正が必要 （実装に影響する可能性がある）	22件
設計書の修正が望ましい （実装に影響する可能性は低いが、修正した方が設計書を理解しやすい）	13件
設計書の修正は不要（※）	20件
合計	55件

（※ 実験者の誤解による指摘や、業務への影響がない指摘）

実際に行われた開発との関係

SEC
Software Engineering
for Mo･No･Zu･Ku･Ri

「修正が必要」と評価された指摘事項22件の内訳

実際の開発における発見時期	件数
後工程（実装・テスト）において発見されていた	13件
実際の開発では指摘されていない （ただし、開発関係者の間では、共通ルールとして徹底されていたため、問題とならなかった）	9件

- 形式手法を使うことにより、従来は実装・テストで発見されていた問題を、設計段階で発見することが可能

形式手法の作業内容と工数の計測

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

- 形式手法を適用する作業を、3段階に分類
  - ① 設計書の読解と形式化する情報の抽出
  - ② 形式記述の作成
  - ③ 形式記述の検証

  （形式手法によっては、作業をさらに細分化）

- 上記作業毎に、かかった工数と、検出した指摘事項を記録

作業内容と指摘事項の関係

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

作業内容と指摘件数の関係

設計書の修正必要性	作業内容			
	文書読解と情報抽出	形式記述の作成	形式記述の検証	合計
修正が必要	1件	19件	2件	22件
修正が望ましい	4件	6件	3件	13件
修正は不要	4件	11件	5件	20件
合計	9件	36件	10件	55件

- **形式記述の作成までで、大半の指摘事項を検出**

（世の中で知られている知見(※)と一致。）

⇒ 厳密な記述が求められるため、設計書を綿密に読み、理解する必要があった。

（※ **S. M. Easterbrook, 他,** Experiences Using Lightweight Formal Methods for Requirements Modeling. : IEEE Transactions on Software Engineering, Special Issue on Formal Methods in Software Practice, vol. 24, (1), 1988.）

作業に要した工数

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

実験チーム	全体工数（人時）	形式化対象の設計書ページ数（頁）	作業効率（人時/頁）	形式記述規模（行）
Event-B（1）	107.5	110	0.98	1,084
Event-B（2）	64.5	106	0.61	443
Event-B（3）	90.0	49	1.84	425
SPIN	44.5	151	0.29	724
VDM++	254.0	300	0.85	10,866

『ソフトウェア開発データ白書2010-2011』 p. 209
「図表8-3-1 ページあたりの基本設計レビュー実績工数の基本統計量(新規開発)」

N	最小	P25	中央	P75	最大	平均	標準偏差
43	0.018	0.065	0.223	1.166	120.267	12.489	30.260

75%のプロジェクトにほぼ収まる。

- 従来のレビューと大差ない作業効率で、形式手法による設計書の検査ができた

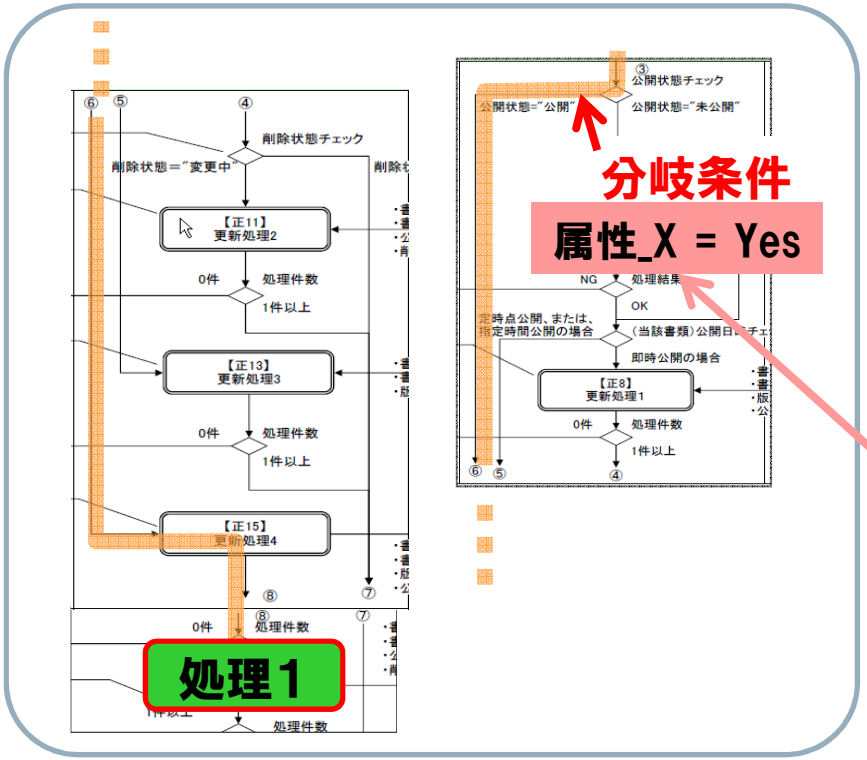
実験で検出された指摘事項の例（1）

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

フローチャート　⇔　画面アクション明細
の間で整合しない記述を発見

（※　実際の開発では、設計書を修正済み）

フローチャート



画面アクション明細

処理1 のアクション明細	
アクション	属性_A を bに更新 属性_B を zに更新
実行条件	属性_A = a　AND 属性_X = No　AND ・・・・

処理を実行する条件が
不一致

実験で検出された指摘事項の例（2）

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

曖昧な記述を発見

画面アクション明細

アクション明細	
アクション	・・・・
実行条件	アクセス制御処理（ID: F0001）の結果がNGの場合

アクセス制御処理設計書

処理定義	
処理ID	F0001
入力	・・・・
出力	○、×、－

どれが「NG」に該当するかが曖昧

「NGの場合」という表現を形式記述しようとして、発見

実験で検出された指摘事項の例（3）

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

記述間違いを発見

（※　実際の開発では、設計書を修正済み）

画面アクション明細

アクション明細	
アクション	・・・・
実行条件	削除状態 ≠ 非公開

正しくは、
公開状態 ≠ 非公開
だった

状態の定義

状態種類	値
削除状態	なし
	変更中
	削除済
公開状態	○○○
	△△△
	非公開

削除状態には
「非公開」という値は無し

「削除状態 ≠ 非公開」と書いた形式記述に対して、
検証ツールが「型の不一致」を検出

まとめ

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

- **外部設計書の検査に形式手法を適用した場合の効果を確認**
  - 従来のレビューでは発見されず、後工程（実装・テスト）で修正されていた問題を発見できた
  - 形式手法により、設計品質を高められること、後工程の修正コストを削減できることの可能性を示すことができた
- **形式手法の適用に要する作業工数を確認**
  - 従来のレビューとほぼ同じ作業効率で、形式手法による設計書の検査ができた
- **参加者（ユーザ企業の委員）のコメント**
  - 「これほどの指摘が出るとは思っていなかった」
  - 「設計書の品質を向上させるための方策の一つとして、従来のレビューに加えて形式手法を採用することは十分可能」

ご清聴ありがとうございました

今回の実験に関する詳しい報告書を、IPA SECのホームページで公開しています。

http://sec.ipa.go.jp/reports/20120420.html