

ランダムテストにおける 問題改善手法の事例報告

株式会社 D T S
清水有樹 田処将司
2012年10月26日

プロジェクト紹介

- 開発対象：Mobile FeliCa IC チップ Firmware
- 開発形態：派生開発
- 開発言語：C言語、アセンブリ
- 開発規模：約128,000 Step(実行コード)
- 開発人員：44名
- 開発期間：2011/10 ~

Mobile FeliCa について

- Mobile FeliCaとは？

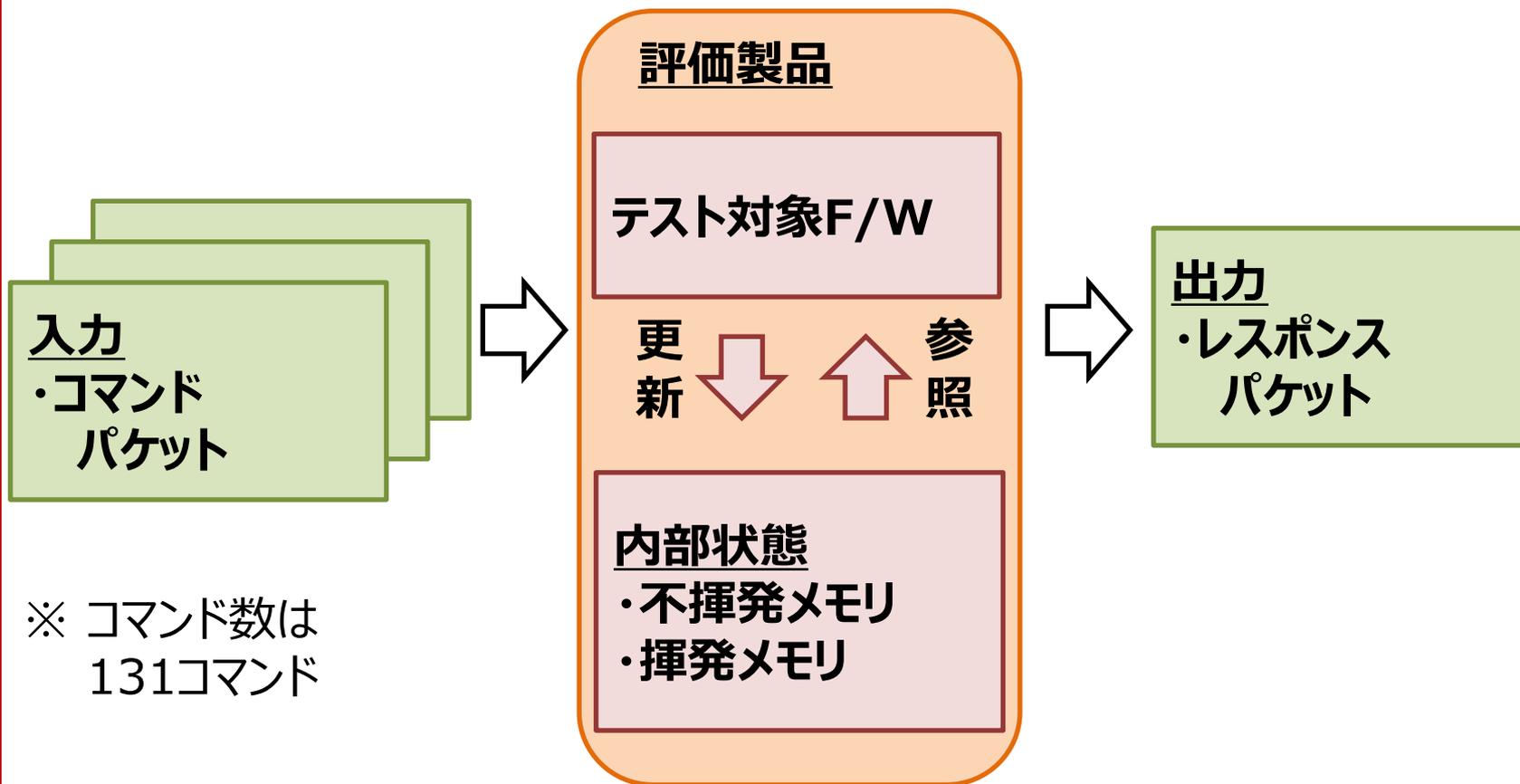
一枚のカードで複数の機能を持つことが出来るFeliCa技術を携帯電話組み込み用ICチップで実現したもの

- 利用シーン

電子マネーや定期券など、生活基盤を支える機能を提供

=>システムやサービスにトラブルが発生した場合の社会的影響は計り知れない

テスト対象



※ コマンド数は
131コマンド

品質に対する取り組み

- 実施しているテスト

- 機能テスト
- ユースケース、シナリオテスト
- インターフェーステスト
- タイミングテスト
- etc.

=> 様々な観点でテストを行っているが、
検出すべき欠陥がすべて取り除けるのか？
テスト間の隙間で漏れている観点がないか？

想定テストケース数

- 想定テストケース数（概算）

入力の組み合わせパターン数を下記の式で算出

$$\begin{aligned} & \text{入力種類} * (\text{水準} ^ \text{因子数}) \\ & = 131 * (4 ^ 8) = 8585216 \text{ 通り} \end{aligned}$$

更に内部状態（揮発メモリ・不揮発メモリ）や
コマンドの組み合わせが絡むと
テストケース数が爆発してしまう

=> 膨大なテストケースを全評価するのは不可能

ランダムテストの導入

- 膨大なテストケースを全評価するのは不可能
- 2因子間の組み合わせ
=> All-Pair法を使って別テストで実施
- 多因子間の組み合わせ
網羅的にバリエーションを持たせて実施できないか？
=> ランダムテストを導入

ランダムテストの目的

- パラメータの組み合わせの充実

他のテストだけでは網羅できていない多因子のパラメータを組み合わせたテストを実施

- 他のテストの十分性の確認

ランダムテストの実施で検出した欠陥を分析し、他のテストで検出されるべき欠陥が検出されないことを確認することで、他のテストの十分性や品質を確認する

一般的なランダムテストの課題

- テスト実施の効率性

無作為にテストの入力を作成すると下記のようなテストケースが出現してしまう

- 重複したテストケース
- 実施不可能なテストケース

=> 期待するテスト空間をテストし終わるまでに時間を要する

一般的なランダムテストの課題

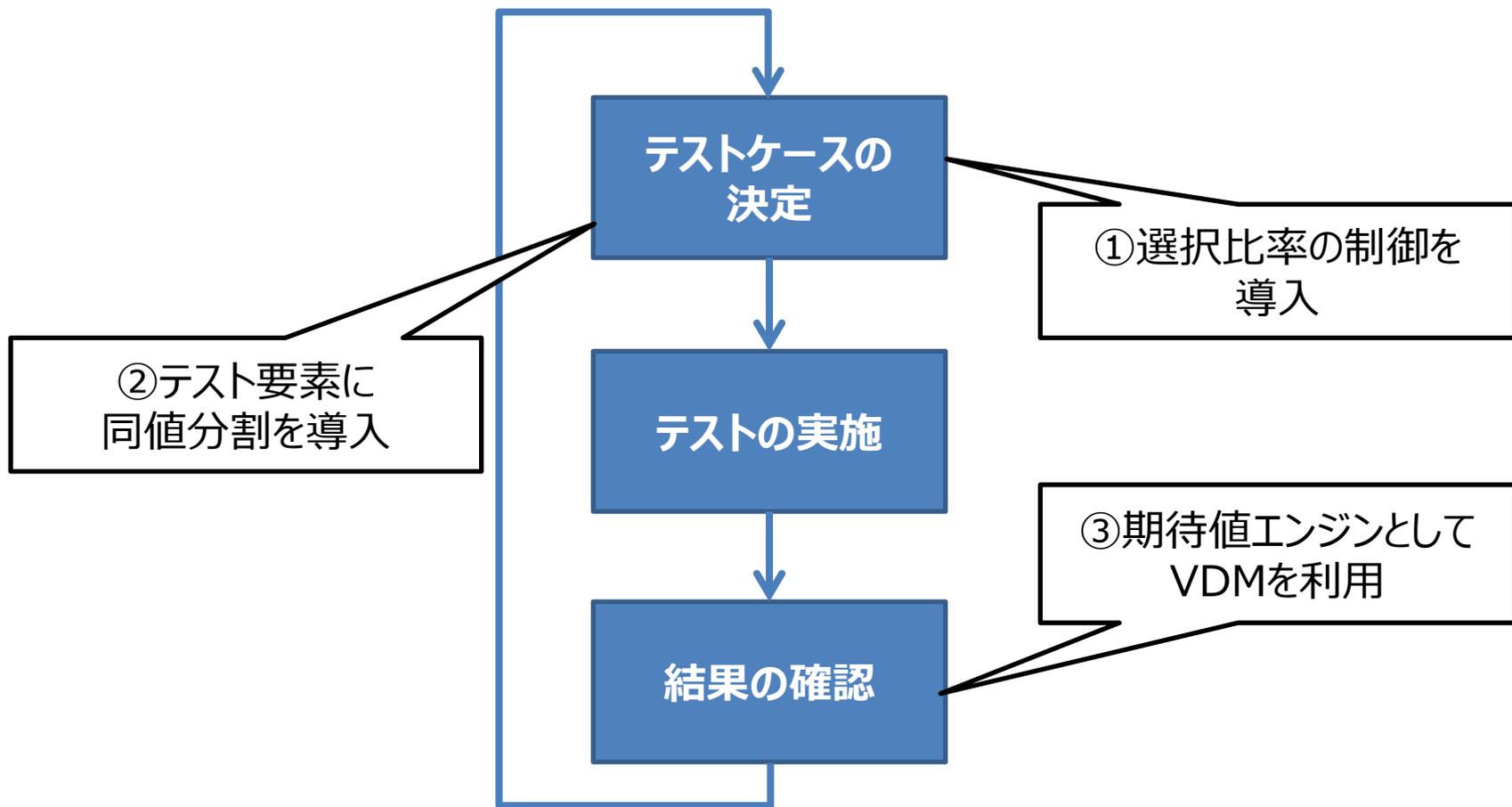
- 結果確認の困難性

無作為にテストの入力を作成し、テストを実施すると、どのような出力になるか不明

=>テスト結果の期待値の作成が困難になり、確認内容が限定的になる

実施した改善策

<ランダムテストのサイクル>



① 選択比率の制御

<改善内容>

- テストケースやテスト種類の選択比率の制御
 - 状態に対応した入力の選択
 - 正常系・異常系の選択

<目的>

- 事前の条件ではじかれないように！
- 重複しないように！ 偏らないように！
- 複雑な条件が評価可能なように！
- 目的に合わせてテストが行えるように！

① 選択比率の制御

<具体例>

テーブルによるコマンド選択

	状態1	ア	ア	ア	イ	イ	イ
	状態2	ウ	エ	オ	ウ	エ	オ
コマンド 選択比率	コマンドA	12	4	12	1	12	17
	コマンドB	13	13	17	3	21	2
	コマンドC	25	21	32	2	14	12
	コマンドD	2	32	27	2	3	4
	コマンドE	4	24	10	39	26	24
	コマンドF	44	6	2	53	24	41
合計		100	100	100	100	100	100

②同値分割の導入

<改善内容>

コマンドの作成にて、要素ごとに同値分割を導入

<目的>

- 効率的に作成できるように！
- ランダム性を活かせるように！
- 実施したテストケースが把握しやすいように！

②同値分割の導入

<具体例>

- 正常系の作成
 - 全要素で同値分割を行い、その時点の内部状態で作成可能な値で組み合わせを行う



- 異常系の作成
 - 正常系を作成した後に 1 箇所または複数箇所を不正な値に置き換える

③期待値エンジン

<改善内容>

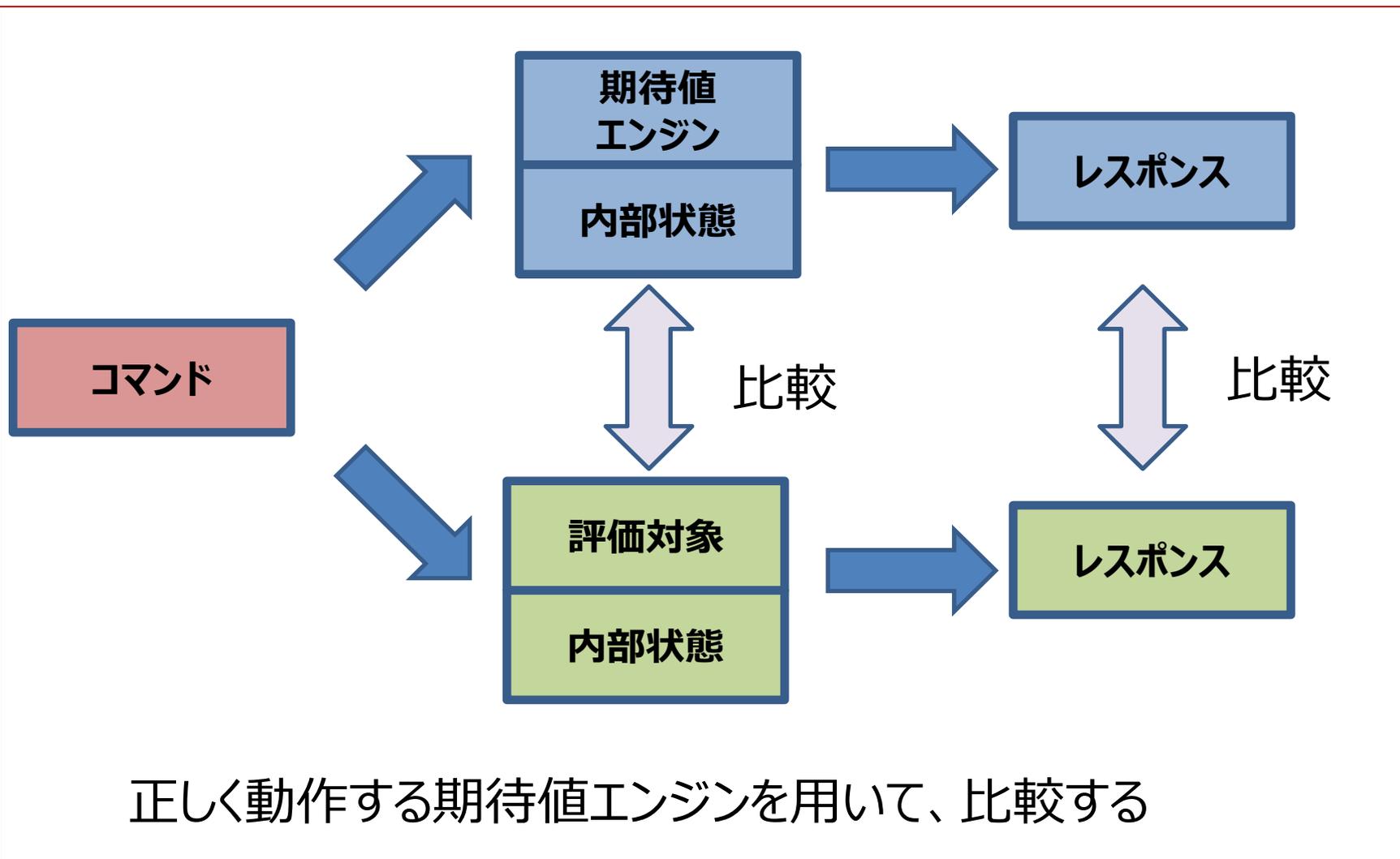
様々な入力に対応した期待値の作成が正確に行える期待値エンジンの導入

- 形式仕様記述言語 VDM++

<目的>

- 正しい期待値が生成できるように！
- 結果の確認を自動化できるように！
- 細部まで結果の確認が行えるように！

③期待値エンジン



正しく動作する期待値エンジンを用いて、比較する

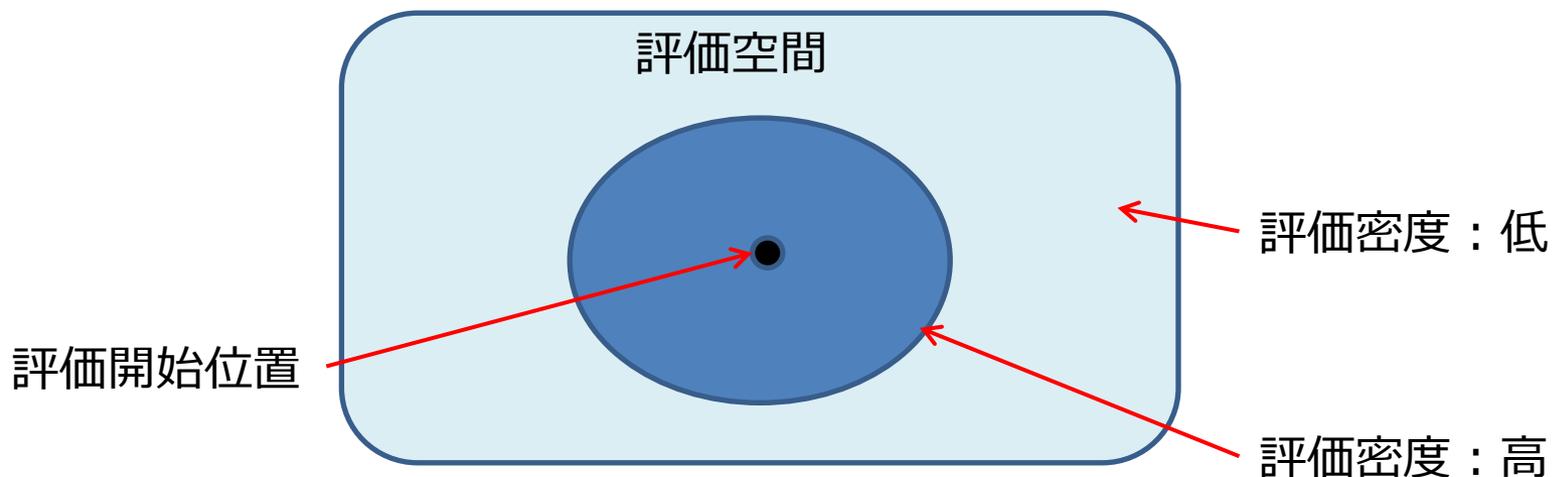
その他の改善

- 評価開始初期状態の制御

初期状態に近い条件 => 評価される可能性が高い

初期状態から遠い条件 => 評価される可能性が低い

評価開始初期状態を制御することで、満遍なく評価を実施



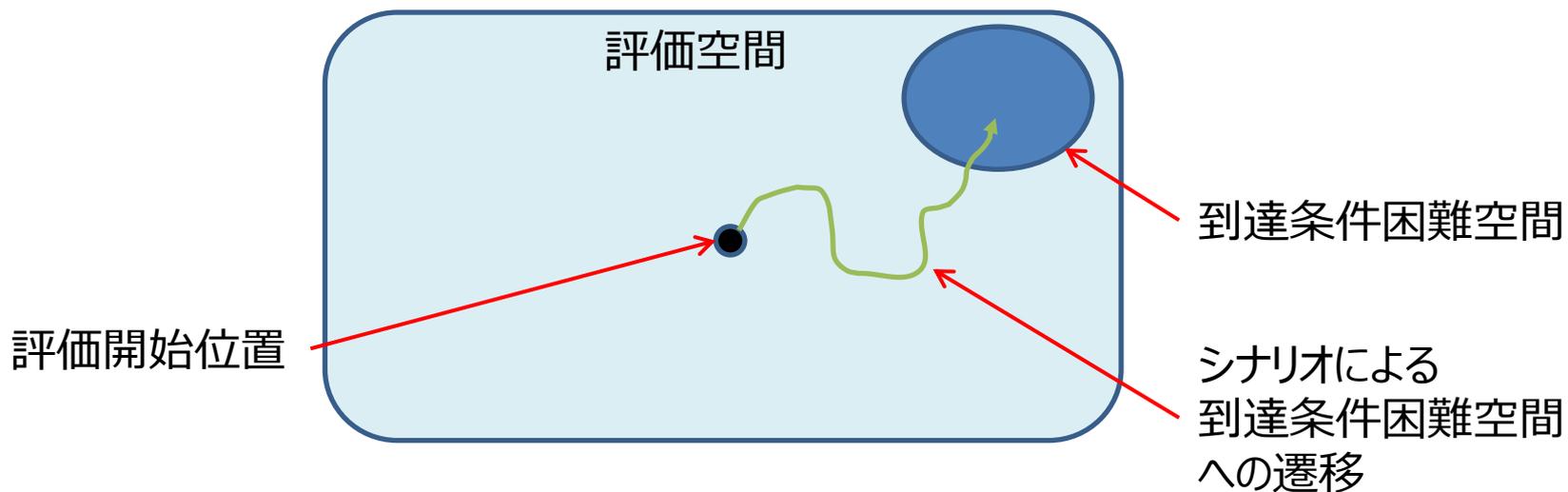
その他の改善

- シナリオパターンの導入

複雑な組み合わせが実行されないと到達しない条件

=> 評価される可能性が低い

複数コマンドをシナリオとして登録し、評価を実施



テスト実施条件と方法

<テスト実施方法>

1週間の走行を複数台で各イテレートで実施

<テスト開始の条件>

- ・2因子間網羅のテストが実施済みであること
- ・運用で想定されるシーケンステストが実施済みであること

<テスト終了の条件>

- ・前後 2 コマンドの組み合わせがすべて発生していること
- ・同値分割したパケット要素の全同値が出現していること

テスト実施結果

- テスト実施

のべ走行時間 5500 時間

実施件数 4883094 件

検出欠陥数 13 件

- 欠陥の内訳

欠陥の分類	件数
2因子組み合わせ	1 件
多因子組み合わせ	4 件
入力間組み合わせ	4 件
その他	4 件

結果

- ランダムテストの目的に対する結果
 - パラメータの組み合わせの充実
 - 多因子組み合わせの欠陥が検出できており、組み合わせの充実させる目的は満たせた
 - 他のテストの十分性の確認
 - 2因子間組み合わせで欠陥が1件発生したが、2因子間網羅のテストの評価拡充のインプットとすることができた

結果

- ランダムテストの課題に対する結果
 - テスト実施の効率性
 - 1環境で1週間の走行で、約15万件のテストが実施できており、実施時間の効率性は十分に図れた
 - 重複や実施不可をテーブルや比率で事前に省くことで、全コマンドの組み合わせと全同値の出現を比較的短時間で確認することができた
 - 結果確認の困難性
 - 期待値に仕様書として記述しているVDMを用いることで正確な期待値を自動で確認することができた

所感

- ランダムテストと他テストの関係
 - テストの品質確保はレビューなどを実施して確保することが多いが、すべてのテストが間違いなく行えている保証はない
 - テストを間引くこと、重複をなくすことも重要であるが、別観点のテスト同士が補完しあうことも重要
- ランダムテストと仕様・設計・実装の関係
 - 設計時にテストを意識するよう働きかけることはテスターとして重要
 - 仕様で形式仕様などを作成しているのであれば、テストへの活用も検討すると良い

ご清聴、ありがとうございました

株式会社D T S
I C T事業本部 I C T 第三部
清水 有樹
E-Mail : yu-shimizu@dts.co.jp
田处 将司
E-Mail : m-tadokoro@dts.co.jp

