

ソフトウェアテスト技法の習得と 実務への適用

～ 闇雲なテストからの脱却 ～

- | | |
|-----|-----------------------------|
| 主査 | : 奥村 有紀子(有限会社デバッグ工学研究所) |
| 副主査 | : 秋山 浩一(富士ゼロックス株式会社) |
| 副主査 | : 堀田 文明(有限会社デバッグ工学研究所) |
| 研究員 | : 佐藤 佳延(株式会社キャノンITソリューションズ) |
| | 古庄 一磨(キャノンファインテック株式会社) |
| | 南雲 博人(株式会社ニコンシステム) |
| | 豊福 暁久(株式会社リンクレア) |
| | 青木 滋(NTTデータシステム技術株式会社) |

目次

- * **研究課題**
- * **習得したソフトウェア技法**
- * **ソフトウェア技法の適用**
 - * **デシジョンテーブル**
 - * **ALL-*Pair*法**
- * **まとめ**

研究課題

- 論理的根拠を持たない独自方法でのテスト
- 担当者のスキルや経験に依存したテスト



- 同じ範囲を何度もテストすることになる無駄
- 未テストの範囲が発生する等のムラが発生
- 品質の確保ができない

研究課題

ソフトウェアテスト 技法の適用

- 目的を明確にしたソフトウェアテスト計画の立案
- テスト実施範囲の把握が容易

- 
- テスト効率の上昇
 - テスト漏れ、抜けの防止
 - 品質の確保

習得したソフトウェア技法(1)

技法	概要
同値分割法	単体テストにて、入力値を同値クラスに分割し代表値を用いる技法
境界値分析法	単体テストにて、入力値に同値クラスの境界値を用いる技法
CFD法 デシジョンテーブル	結合テストにて、複雑な内部仕様を流れ図で表しデシジョンテーブルで入力と出力の組合せを示す技法
直交表 HAYST法	組合せテストにて、実験計画法で用いられる直交表の性質を利用し2因子間の組合せを網羅したテストケースを設計する技法
All-Pair法	組合せテストにて、任意の2因子間の全水準の組合わせが100%となるテストケースを設計する技法
状態遷移テスト	結合テストにて、テストすべき状態とイベントを表にし、テストケースを設計する技法
マインドマップ	中央に据えたキーワードから、放射状にキーワードを広げ、思考を視覚的に表現する記法。テスト観点を導き出すために利用

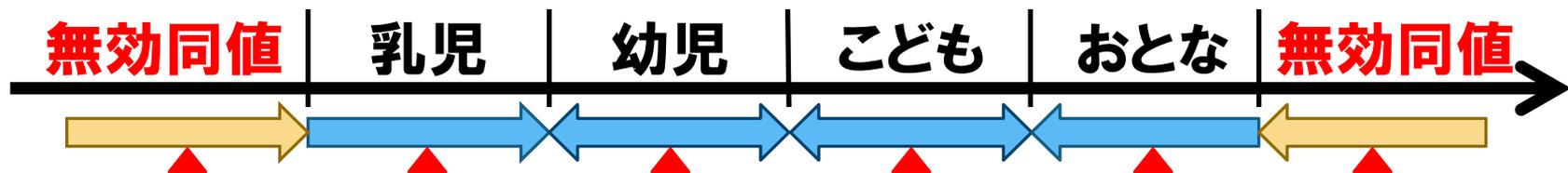
習得したソフトウェア技法(2)

【同値分割法】

- ・同値分割法とは、同値または同値クラスと呼ぶ入力値を**同一特性の部分集合に分割し、代表値入力が出力結果をテストする技法。**

JR の料金体系

- ・おとな: 12 歳以上
- ・子ども: 6 歳 ~ 12 歳未満
- ・幼児: 1 歳 ~ 6 歳未満
- ・乳児: 1 歳未満



習得したソフトウェア技法(2)

【同値分割法】

[適用メリット]

- ・プログラムの処理に影響する入力を部分集合することにより**テストケースを削減**する

[気づき]

- ・最も認知されているテスト技法の一つであり、必須といえる
- ・過去のテストケースを振り返ると、前提となる同値分割を正しく行わないまま適切でない境界値でテストを実施している場合もあった

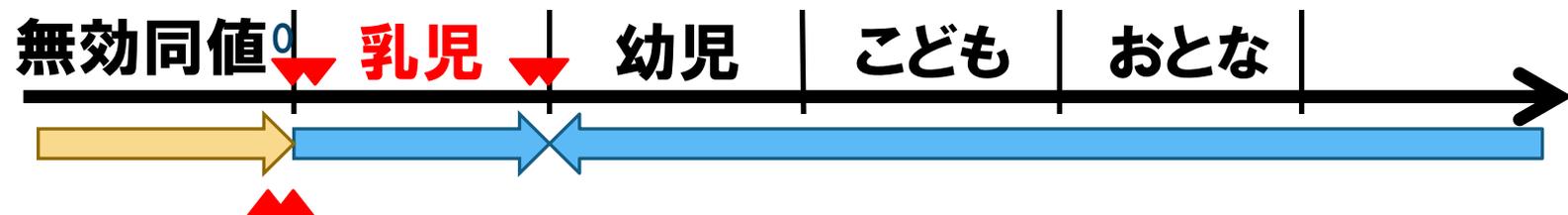
習得したソフトウェア技法(3)

【境界値分析法】

- ・同値分割法とは、同値または同値クラスと呼ぶ入力値を**同一特性の部分集合に分割し、代表値入力が出力結果をテストする技法。**

JR の料金体系

『乳児』クラスのみを考慮した境界値は



習得したソフトウェア技法(3)

【境界値分析法】

[適用メリット]

- ・ **障害を作り込む可能性が高い境界値を確認**できる

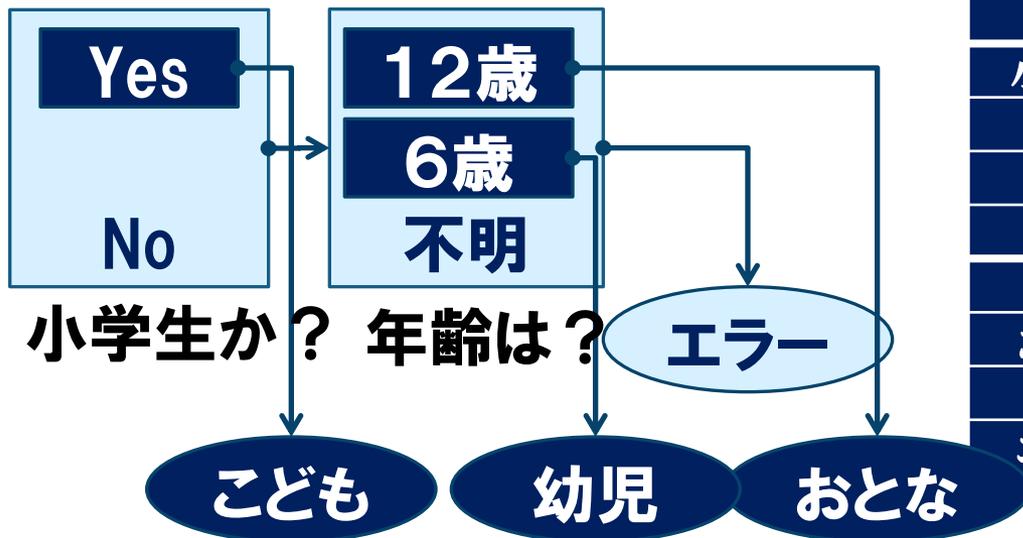
[気づき]

- ・ 過去のテストケースを振り返ると、有効側の境界値のみ確認しており、無効側の境界値を確認していない場合もあった

習得したソフトウェア技法(4)

【CFD法】【デシジョンテーブル】

- CFD法(Cause Flow Diagram)とは、複雑な内部仕様を流れ図で表す技法。
- デシジョンテーブルとは入力と原因、及び対応する出力と結果の組合せを示す技法。



料金体系	ルール 1	ルール 2	ルール 3	ルール 4
小学生	T	F	F	F
12歳	-	T	F	F
6歳	-	-	T	F
不明	-	-	-	T
こども	○			
おとな		○		
幼児			○	
エラー				○

習得したソフトウェア技法(4)

【CFD法】【デシジョンテーブル】

[適用メリット]

- ・結果を有効系、またはエラー処理等の無効系に分割し、**テスト実施の優先順位をつけることができる**

[気づき]

- ・各処理間の関係も見えるため、仕様として記述されていなかった曖昧な部分に気づく
- ・条件が複雑で判断が難しい仕様もデシジョンテーブルを適応させることによって、わかりやすく整理することができる

習得したソフトウェア技法(5)

【直交表】【HAYST法】

- ・直交表とは、任意の2因子(列)にて、とり得る**選択肢(水準)**の**全ての組合せが同数回ずつ現れる**という性質を持ったテストの為の割り付け表
- ・HAYST法とは、直交表の性質を利用して**2因子間の組合せを網羅したテストケースを設計する技法**

	因子A	因子B	因子C
ケース1	0	0	0
ケース2	0	1	1
ケース3	1	0	1
ケース4	1	1	0

習得したソフトウェア技法(5)

【直交表】【HAYST法】

[適用メリット]

- ・機能と機能間に仕様上で関係性がないとされている場合の組合せに有効である

[気づき]

- ・因子と水準さえ決まっていればあとは技法のルールに従ってテストケースを作成すればよい
- ・因子・水準の選定に関してはテスト要求分析やテストアーキテクチャ設計を注意して行わないと無駄なテストケースが出てしまうので注意が必要

習得したソフトウェア技法(6)

【All-Pair法】

- ・任意の**2因子間の全水準の組合せが100%となるテストケース**を作成する技法
- ・HAYST法との違いは、任意の2因子間の「**同数回**」水準組合せを**All-Pair法では「同数回」という条件を外す点**

	因子A	因子B	因子C
ケース1	0	0	0
ケース2	1	0	1
ケース3	0	1	1
ケース4	1	1	0
ケース5	0	0	2
ケース6	1	1	2

※因子Aと因子Bにおいて、
(0.0) (1.1) は2回出現するが、
(1.0) (0.1) は1回のみ出現
する

習得したソフトウェア技法(6)

【All-Pair法】

[適用メリット]

- ・任意の**2因子間の水準の組合せに着目し**、比較的簡単にテストする組合せを作成できる

[気づき]

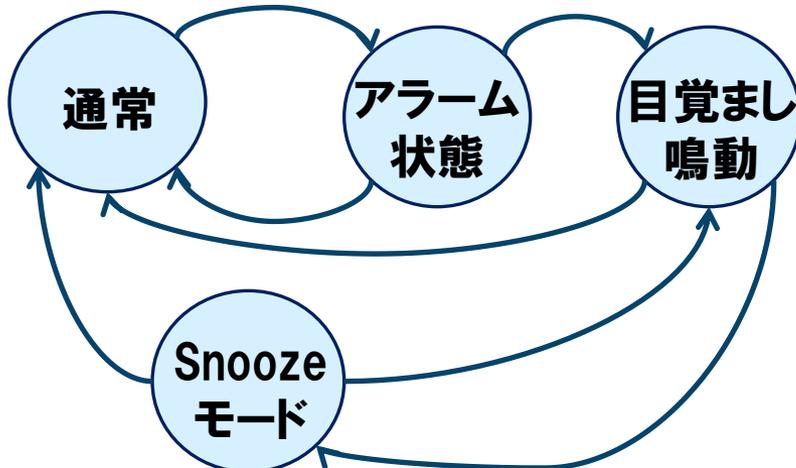
- ・組合せ作成ツール等もあり、実作業への適用が容易である
- ・禁則を有効に使うことによってテストケースを減らすことができる

習得したソフトウェア技法(7)

【状態遷移テスト】

・状態と遷移パスを洗い出し状態遷移図を作成し、そこから状態遷移表にイベントをマトリクスに表現した表を作成する。状態遷移表をもとに**状態モデル全体を網羅的にテスト**する技法。

状態遷移図



第5分科会(Aグループ)

状態遷移表

	アラーム セット	セット 時刻	Snooze ボタン	5分経 過	アラーム 解除
通常	アラーム 状態	-	-	-	-
アラーム 状態	-	目覚まし 鳴動	-	-	通常
目覚まし 鳴動	-	-	Snooze モード	-	通常
Snooze モード	-	-	-	目覚まし 鳴動	通常

習得したソフトウェア技法(7)

【状態遷移テスト】

[適用メリット]

- ・複数の状態が遷移する様子表現する状態遷移図を作成することにより**仕様の不備や誤りを検出できる**

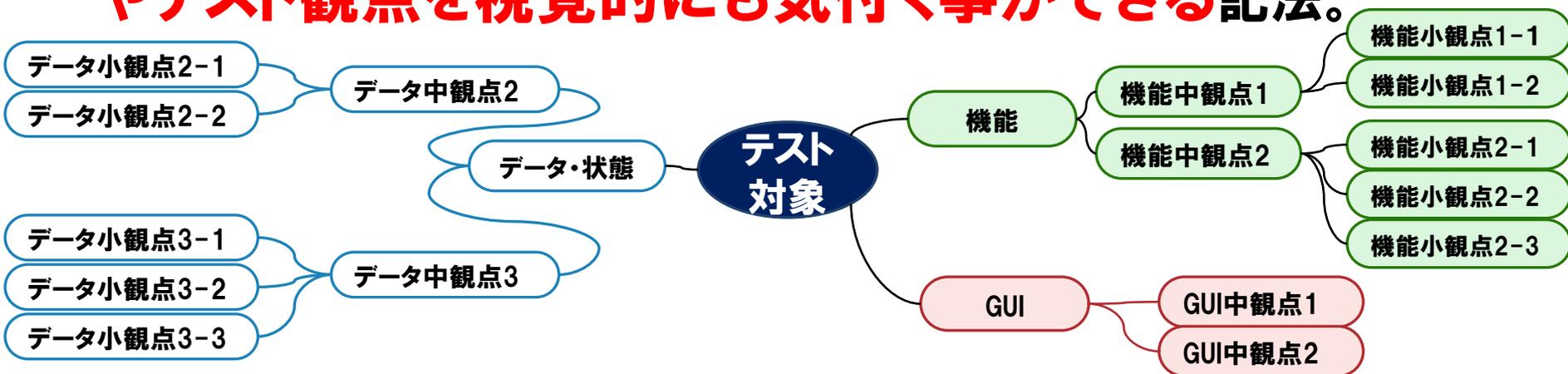
[気づき]

- ・過去のテストケースを振り返ると、各状態に遷移しているか確認する程度のもので、状態ごとのパスに注目した評価はしていない場合もあった

習得したソフトウェア記法(8)

【マインドマップ】

・仕様書から「仕様の整理」、「テスト観点の抽出」を行う際に、テスト設計時のベースとなるドキュメントを読み解く際3色ボールペンを使って仕様を整理する。次にテスト観点の抽出を行う際に色分けしたマインドマップを作成することで、**見落としがちな仕様やテスト観点を視覚的にも気付く事ができる記法。**



習得したソフトウェア記法(8)

【マインドマップ】

[適用メリット]

- ・**思考の可視化に優れて**おり、テストの観点(機能、非機能等のさまざまなテスト目的の概念)を導き出すことに有効

[気づき]

- ・書き方によっては、発想が広がり難いものになってしまうため、書き方や議論の進め方にも経験が必要である。
- ・テスト観点の作成にも利用可能だと初めて知った。テストケースの網羅ではなく、全体像を視覚的に把握でき、関連している要素や観点に思い至りやすいという利点がある。

ソフトウェア技法の適用(1)

デシジョンテーブル

A社の悩み

2機能以上の評価方法が**不明確**

担当者ごとに**ばらつき大**

**評価漏れ発生
バグ流出**

テスト技法未導入
経験やスキルで
テストケース作成

市場問題

ソフトウェア技法の適用(1)

デシジョンテーブル

A社の悩み

2機能以上の評価方法が**不明確**

担当者ごとに**ばらつき大**



**デシジョンテーブルを適用して
複数機能評価方法の明確化！！**
※今事例では6機能間にDT適用

ソフトウェア技法の適用(1)

デシジョンテーブル

【デシジョンテーブル適用手順】

(1) 仕様の整理と条件の洗い出し

前準備として、複数機能に関係する入力値、動作状態等を表にまとめた。

機能	入力条件	環境1	環境2	処理内容	目的	備考
機能1	全適応	全適応	全適応	処理内容1	目的1	機能2と同時設定不可
機能2	入力1、入力3があった場合	全適応	全適応	処理内容2	目的2	機能1と同時設定不可
機能3	全適応	全適応	高温の場合	処理内容3	目的3	特になし
機能4	入力1、入力2、入力4があった場合	全適応	全適応	処理内容4	目的4	特になし
機能5	入力1、入力3があった場合	高温の場合	全適応	処理内容5	目的5	特になし
機能6	全適応	全適応	全適応	処理内容6	目的6	特になし

	機能1	機能2	機能3	機能4	機能5	機能6
機能1	■	○	△	○	×	○
機能2	×	■	△	△	×	○
機能3	△	△	■	△	○	△
機能4	×	△	△	■	×	△
機能5	○	○	×	○	■	○
機能6	×	×	△	△	×	■

○: 横が優先
 ×: 縦軸が優先
 △: 重複可能

◆各機能の仕様まとめ例

◆優先順位まとめ例

仕様の抜け漏れ防止のため
関係者にレビュー必須

ソフトウェア技法の適用(1)

デシジョンテーブル

【デシジョンテーブル適用手順】

(2) デシジョンテーブルを使って全組合せの書き出し

条件指定部に選択した条件を書き出し、各条件全組合せを入力した。

※今回の条件の場合、768組合せとなった。

No	条件												処理												
	機能						入力				環境1		環境2		機能1	機能2	機能3	機能4	機能5	機能6	通常動作	設定不可			
	機能1	機能2	機能3	機能4	機能5	機能6	入力1	入力2	入力3	入力4	低温	高温	低温	高温											
1	Y		Y				Y						Y		Y	Y		Y							
2	Y			Y			Y						Y		Y	Y									
3	Y				Y		Y						Y		Y					Y					
4	Y					Y	Y						Y		Y	Y									
5	Y		Y	Y			Y						Y		Y	Y		Y							
6	Y			Y	Y		Y						Y		Y					Y					
7	Y				Y	Y	Y						Y		Y					Y					
8	Y		Y			Y	Y						Y		Y	Y		Y							
...																									
767			Y		Y	Y						Y	Y		Y										Y
768			Y	Y	Y	Y						Y	Y		Y										Y

仕様のまとめ表や優先順位表を用いて結果指定部に記入していく。

ソフトウェア技法の適用(1)

デシジョンテーブル

【デシジョンテーブル適用手順】

(3) 評価への適用

評価をスムーズに進めヒューマンエラーを防止するために、処理結果が同じ内容の項目をまとめ整理

No	条件										処理											
	機能						入力				環境1		環境2		機能①	機能②	機能③	機能④	機能⑤	機能⑥	通常動作	
	機能①	機能②	機能③	機能④	機能⑤	機能⑥	入力①	入力②	入力③	入力④	環境①	環境②	環境①	環境②								
1	Y			Y			Y					Y		Y								
2	Y				Y		Y					Y		Y								
3	Y					Y	Y					Y		Y								
4	Y			Y	Y	Y	Y					Y		Y								
5		Y		Y			Y					Y		Y							Y	
6		Y			Y		Y					Y		Y							Y	
7		Y				Y	Y					Y		Y							Y	
8		Y		Y	Y	Y	Y					Y		Y							Y	

ソフトウェア技法の適用(1)

デシジョンテーブル

デシジョンテーブル適用の効果

- 一定のルールに従い複数機能の組合せに対するテストケースが作成できた
- 仕様の整理とレビューからデシジョンテーブルを適用し、漏れなくかつ分かりやすく表現できた

デシジョンテーブル適用後の課題

- 仕様整理の実施とその方法についてルール化
- デシジョンテーブルで作成したテストケースの合理的削減

ソフトウェア技法の適用(2)

All-Pair法

B社の悩み

評価方法の社内標準が**未確立**

評価内容はテストケース作成者のスキルに依存

影響範囲の見極めミスで想定外の機能間に障害が...

社内標準になりうるテスト技法を用いて、想定外の障害を防ぎたい！！

品質問題

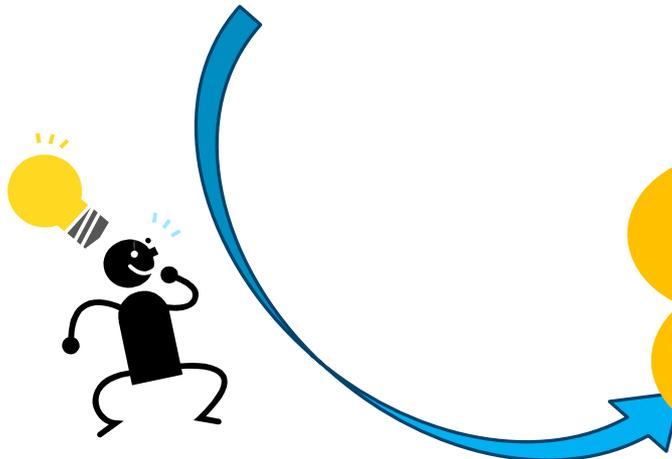
ソフトウェア技法の適用(2)

All-Pair法

B社の悩み

評価方法の社内標準が**未確立**

評価内容はテストケース作成者のスキルに依存



**All-Pair法を適用して
機能間に想定外の障害が
無いか確認！！**

※今事例では21因子・46水準を適用

ソフトウェア技法の適用(2)

All-Pair法

【All-Pair法適用手順】

(1) 設計書から因子や水準を抽出し、ツールに読み込む。なお、ツールはQumiasを採用し、今回は21因子・46水準を適用した。

- 水準2個の因子: 18個
- 水準3個の因子: 2個
- 水準4個の因子: 1個



The screenshot shows the Qumias tool interface in Microsoft Excel. The tool is open, displaying a menu with options: 新規作成 (New), 既存のモデルファイルから作成 (Create from existing model file), and About. Below the menu, a table lists 21 factors and their corresponding levels.

因子	水準
因子1	水準1-1
因子2	水準2-1
因子3	水準3-1
因子4	水準4-1
因子5	水準5-1
因子6	水準6-1
因子7	水準7-1
因子8	水準8-1
因子9	水準9-1
因子10	水準10-1
因子11	水準11-1
因子12	水準12-1
因子13	水準13-1
因子14	水準14-1
因子15	水準15-1
因子16	水準16-1
因子17	水準17-1
因子18	水準18-1
因子19	水準19-1
因子20	水準20-1
因子21	水準21-1
	水準1-2
	水準2-2
	水準3-2
	水準4-2
	水準5-2
	水準6-2
	水準7-2
	水準8-2
	水準9-2
	水準10-2
	水準11-2
	水準12-2
	水準13-2
	水準14-2
	水準15-2
	水準16-2
	水準17-2
	水準18-2
	水準19-2
	水準20-2
	水準21-2
	水準8-3
	水準10-3
	水準13-3
	水準13-4

ソフトウェア技法の適用(2)

All-Pair法

【All-Pair法適用手順】

(2) ツールで網羅度の設定・制約条件の設定・試行回数設定を行う

① 網羅度の設定: 2項目間の網羅(デフォルト値)

② 制約条件の設定: 「if A=1 then B=2 and C=3」といった制約条件6種類

③ 試行回数の設定: 設定なし(デフォルト値)

網羅度設定

全体網羅度の設定
全体で、いくつの項目間で網羅するか指定できます。
 任意の項目数(2~6) 全網羅

指定項目の指定
指定項目の網羅度を上げることができます。
網羅すべき項目を最大10項目まで選択してください。
ただし、全1網羅度以下にはできません。

現在の設定で作成

制約条件編集

ルール名
ルール4

条件(IF)
項目 因子1 が
値 水準1-1
 と等しい場合 と等しくない場合
追加(AND) 追加(OR) 編集 削除

制約(THEN)
項目 因子1 が
値 水準1-1
 と等しい と等しくない
追加(AND) 追加(OR) 編集 削除

条件(IF)
因子7: 水準7-1

制約(THEN)
因子8: not 水準8-3
因子9: not 水準9-2
因子10: not 水準10-3
因子11: not 水準11-2

キャンセル OK

試行回数・シード値

最小テストケース作成
組み合わせの作成
組み合わせ数が最も少ない組み合わせのみを生成

指定回数

シード値の指定
組み合わせを作成する
以前作成した組み合わせのみを生成

シード値

現在の設定で作成

第5分科会(Aグループ)

2013/2/15

ソフトウェア技法の適用(2)

All-Pair法

【All-Pair法適用手順】

(3) ツールを実行し、テストケースを作成する。

実行結果として、入力データ・制約条件一覧・テストケースが1ファイルとして出力される。

The image displays two overlapping Excel spreadsheets. The left spreadsheet, titled '20130215.xlsx', is labeled 'テストケース' (Test Cases) and shows a grid of test cases with columns for factors (因子1-12) and their levels (e.g., 水準1-1, 水準2-1). The right spreadsheet is labeled '制約条件' (Constraints) and lists rules (ルール名) with conditions (条件(IF)) and constraints (制約(THEN)).

ルール名	条件(IF)	制約(THEN)
ルール1	因子7: 水準7-2	因子8: 水準8-3 └ 因子9: 水準9-2 └ 因子10: 水準10-3 └ 因子11: 水準11-2
ルール2	因子16: 水準16-2	因子17: 水準17-2 └ 因子18: 水準18-2
ルール3	因子19: 水準19-2	因子20: 水準20-2 └ 因子21: 水準21-2
ルール4	因子7: 水準7-1	因子8: not 水準8-3 └ 因子9: not 水準9-2 └ 因子10: not 水準10-3 └ 因子11: not 水準11-2
ルール5	因子12: 水準12-1	因子13: not 水準13-4 └ 因子14: not 水準14-2
ルール6	因子12: 水準12-2	因子13: 水準13-4 └ 因子14: 水準14-2

ソフトウェア技法の適用(2)

All-Pair法

All-Pair法試行の効果

- **2因子間の全水準の組合せが100%のテストケースが作成できた**
- **初心者でもツールが直感的に操作でき、容易にテストケースが作成できた**

All-Pair法試行後の課題

- **因子や水準の効果的な抽出方法のガイド作成**
- **他ツール (PictMaster) との、機能や使い易さの比較**

まとめ

様々なテスト技法の理解を深め、一定のルールに従った漏れのないテストケースの作成方法を身につける事ができた

➤ 今後の課題

- 社内普及にあたり、技法適用時のルール明確化
- 効率的な適用方法の調査



ご清聴ありがとうございます