組込みソフトウェア検証における「組合せテスト」の更なる可能性の追求

JaSST '07 Tokyo

2007年1月30日

バルテス株式会社

大阪本社 〒541-0052 大阪市中央区安土町3-5-12
住友生命本町ビル7F

TEL : 06-6267-6500 FAX : 06-6267-6501
URL : http://www.valtes.co.jp/
E-mail : ishihara@valtes.co.jp

技術開発室 石原 一宏

本日の流れ

1　組込みソフトウェアテストに求められる
　「コスト」と「品質」

2　高品質なソフトウェアテスト設計とは

3　モデリング・アプローチ①
　「ユーザーの操作条件を時間軸で組合わせる」

4　モデリング・アプローチ②
　「回避すべきクリティカルな不具合をいかに
　　テストで食い止めるか」

5　ソフトウェアテストの可能性の追求

第1章
組込みソフトウェアテストに求められる「コスト」と「品質」

組込みソフトウェア品質は「テストの品質」にかかっている

VALTES

- **付加価値の要求**
  - →ソフトウェアライン数の増大
  - →テスト工数の爆発的増大
- **派生開発サイクルの短縮**
  - →更なるテスト期間の短縮
- **消費者意識、PL意識の高まり**
  - →安全面、使い勝手、信頼性への高い要求

100%テストすることは不可能

- 「プログラムが全くエラーを持たないと保障するようなテストは出来ない」

- 「プログラムテストの基本的な考え方には、経済的な見方が必要である。徹底的なテストは不可能であるから、その投資に対して最大の効果をもたらすことを目的とすべきである」

G.Myers『ソフトウェア・テストの技法』

そのテストにおいて
何を最優先課題にするのか

優先順位を付け、限られたリソースの中で
検証する必要がある

明確な「品質獲得目標」を持たなくては
ならない

ソフトウェア品質獲得目標の「3タイプ」

VALTES

① 「検出型」：不具合検出

② 「網羅型」：設定範囲内を全網羅

③ 「最善型」：①と②の両方の特性を
すり合わせる

しかし、それだけでは十分とはいえない

更に求められるテストのパフォーマンス

- ユーザーの操作順序によって起こる不具合を可能な限りテストしたい
- 優先順位の高いリスクを効果的に回避したい

「網羅配列表」は確かに便利。だが・・・

- 直交表やAll-Pair法は工数を劇的に削減してくれる強力な「道具」である

  L64直交表は2水準パラメーターを最大63因子扱える。

  総当りなら　　直交表なら

  2^63=

  9,223,372,036,854,775,808（約922京）通り ⇒ 64回

- 何を因子・水準に設定するのかが組合せテスト設計の最大のポイント

第2章
高品質なソフトウェアテスト設計とは

Verification & Validation

VALTES

「Verification」：検証

「正しく製品を作っているか」
仕様書どおりに動作するか？

「Validation」：妥当性確認

「正しい製品を作っているか」
顧客の要求を満足させられる製品であるか？

求められるテスト設計品質

- 「仕様書通り」に動くことを確認する
  「**検証**」(Verification) だけで十分か？

- リスクの重要度に立脚した
  「**妥当性確認**」(Validation) も必要

- ユーザーの使用する操作範囲を可能な限り
  シミュレートすることも求められる

高品質な組合せテスト設計のポイントとは

Point 1

抜け・漏れなくテスト対象項目を設定する

⇒ 細目一覧の作成

Point 2

効率よくモデリングする

⇒ ①時間軸で組合わせる

⇒ ②リスクベースドテスト

Point 1
テスト対象項目を設定する

自然語で記述された外部仕様から
もれなく項目設定に必要な条件を抽出

- 『テスト項目の抜けを無くす！！』
  ⇒細目一覧の作成
  開発者との共同作業が大切
  「隠れた仕様」を引き出す

- 同値分割（入力/出力、有効/無効などは、全て
  ドキュメントにはなっていない）

- 限界値・境界値分析（制御構造、過去の不具合など
  から初めてわかる境界値もある）

細目一覧(Inventory)の例

VALTES

機能名			戦略		仕様（Verification）			観点（Validity
大項目	中項目	小項目	優先度	検出型	印刷	分類	仕様	
規格対応	可能設定	可能設定			○	印刷設定	カメラ側で指定した1枚もしくは2枚以上の画像をprint	設定項目の重複の有無（カメラと本
					○	部数設定	カメラが保持する全画像のprint	
					○	部数設定	1～９９９部までの部数指定print	
	UI表示	状態通知				表示	pict bridge 接続確立したことをユーザーに告知　（カメラ、プリンタの双方）	データの形式や容量を確認
						表示	プリンタ側エラー発生をユーザーに告知（プリンタのみ）	
						表示	プリンタ側エラー発生をユーザーに詳細情報を告知（カメラのみ）	
						表示	ジョブの進行状況をカメラに通知	
						表示	ジョブが完了したことをカメラに通知	
						表示	記録紙補充、カバー開、紙づまり	
	設定可能機能	DPOF印刷			○	印刷	DPOFでマーキングした画像を印刷する	
		IndexPrint			○	印刷	全画像のindex print	
		日付挿入print			○	印刷	on / off	
		ファイル名挿入print			○	印刷	on / off	
		print解像度指定			○	印刷	normal(600×600) / fine(2400×600)	ファイルの画質の反映
		レイアウト指定			○	レイアウト	1in1 / index	
		記録紙サイズ指定			○	サイズ設定	A4,LETTER,JIS B5,ISO B5,A5,ISO B6,A6,default　→1in1印刷	サイズ誤認識の場合
							A4,LETTER　→Index印刷	
	DSC操作				○	外部機器操作	カメラ側からのプリント中止	
					○	外部機器操作	カメラからのcomtinueの指示で印刷動作を復帰する	
動作条件	接続	接続				接続	Pict bridge対応カメラがUSBケーブルを介して接続されていること パネル表示「camera connected」	
		機能通知					プリント機能仕様書（4印刷方式決定）のグレーアウト部分がカメラ側から設定したとき無視される	
							カメラ側に選択肢が現れない場合はプリンタ側でdefaultが指定される	
	対応ファイル	動作対象				対応ファイル	JPEG	対応ファイル以外を使用した場合に
Image印刷	ファイル名挿入	印刷表示			○	印刷設定	on　サイズ10pt固定、それぞれ1行で表記（用紙はみだしOK）	フォルダ名、階層などの制限
		ファイル名の配置位置			○	印刷設定	対応する画像の左端あわせ / 対応する画像の直下	
		Offの場合			○	印刷設定	off　表記されない	
	日時挿入	印刷表示			○	印刷設定	on　サイズ10pt固定、それぞれ1行で表記（用紙はみだしOK）	フォルダ名、階層などの制限
		Offの場合			○	印刷設定	off　表記されない	
		日時の配置位置			○	印刷設定	対応する画像の左端あわせ / 対応する画像の直下	
	Imageの位置	ファイル名・日時Onの場合			○	印刷設定	画像の縦横サイズ＋表示するファイル名、日時の高さが用紙に収まる最大サイズ	
					○	印刷設定	上記領域を縦横中央に配置する	

Point 2
効率よくモデリングする

抽出された項目を品質獲得目標に照らして、
必要最小限のテスト項目に設定する

そうしないと容易に発散してしまう！

⇒『切り出した項目をモデリングする』

⇒そこでテスト設計の技術と思想性が問われる

第3章

モデリング・アプローチ①

『ユーザーの操作順序を
いかにして組合わせるか』

迫られるユーザー操作順序の組合せテスト

VALTES

- ユーザーの操作順序によって起こる不具合を可能な限りテストしたい
- クレームにつながる不具合をいかに効果的にテストで摘出するか
- ユーザー操作順序のシミュレート

  ⇒状態遷移の『時間軸組合せ』

空間モデルと時間モデル

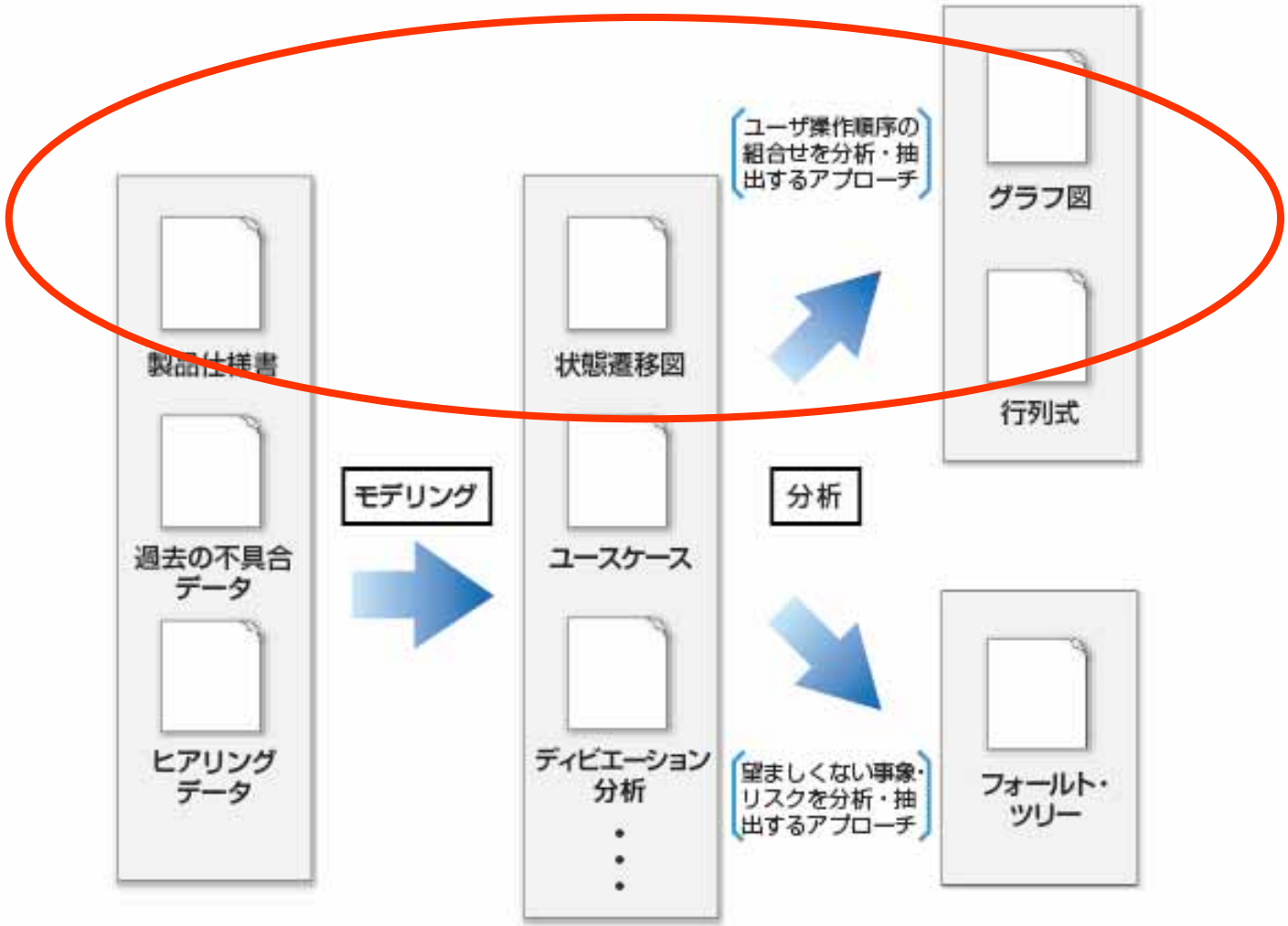
空間モデル（LSM：Logical Space Model）

- 『水平の組合せ』
- 「集合の構造」モデル。相互に交差するサブセットを「集合の直積構造」としてハンドリングする。
  適用例：ユーザー設定、DIP設定の組合せ

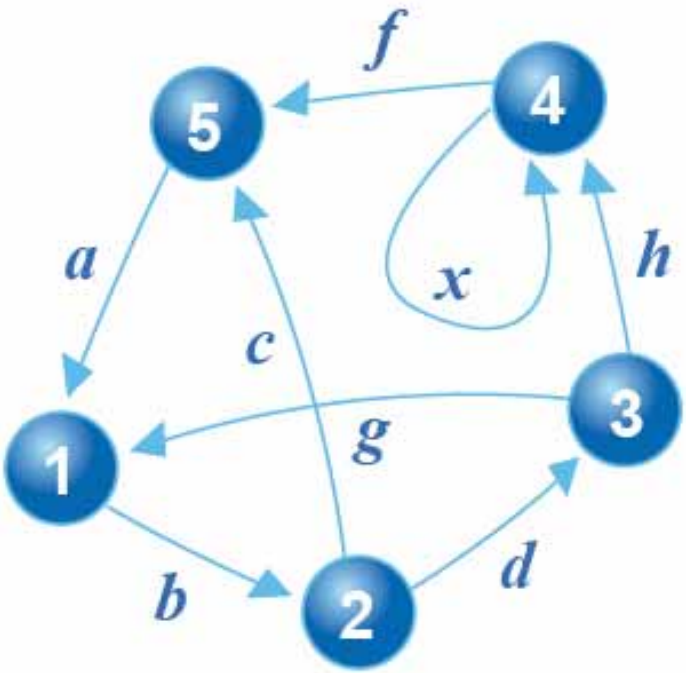
時間モデル（FSM：Finite State Machine Model）

- 『垂直の組合せ』
- 「有限順序機械」モデル。時間軸の順序組合せをハンドリングする。⇒状態遷移図の簡略化
  適用例：ユーザー操作順序組合せ

テスト設計作業の流れ



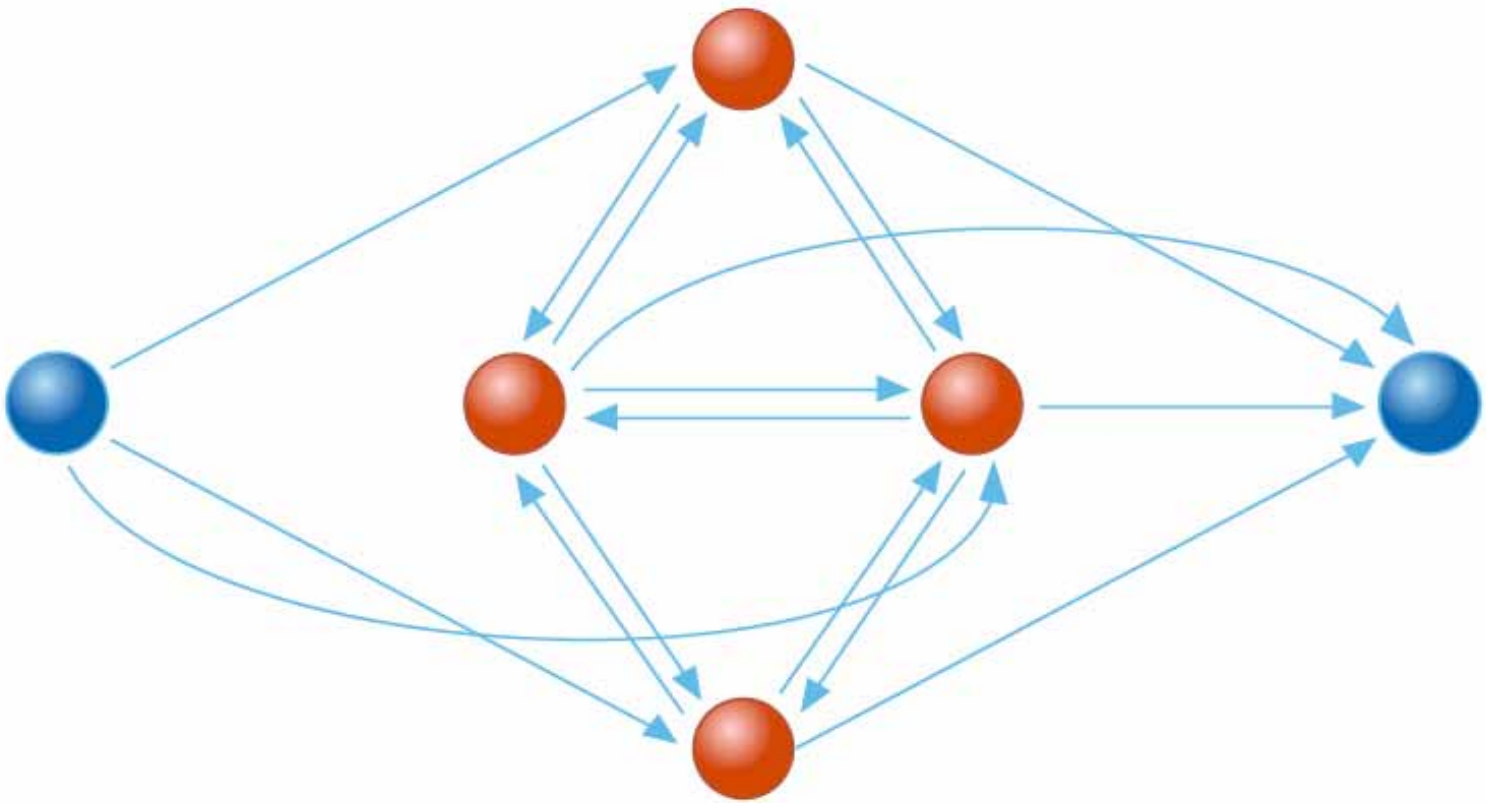
モデリングした状態遷移図を
グラフ理論によって行列式に変換する



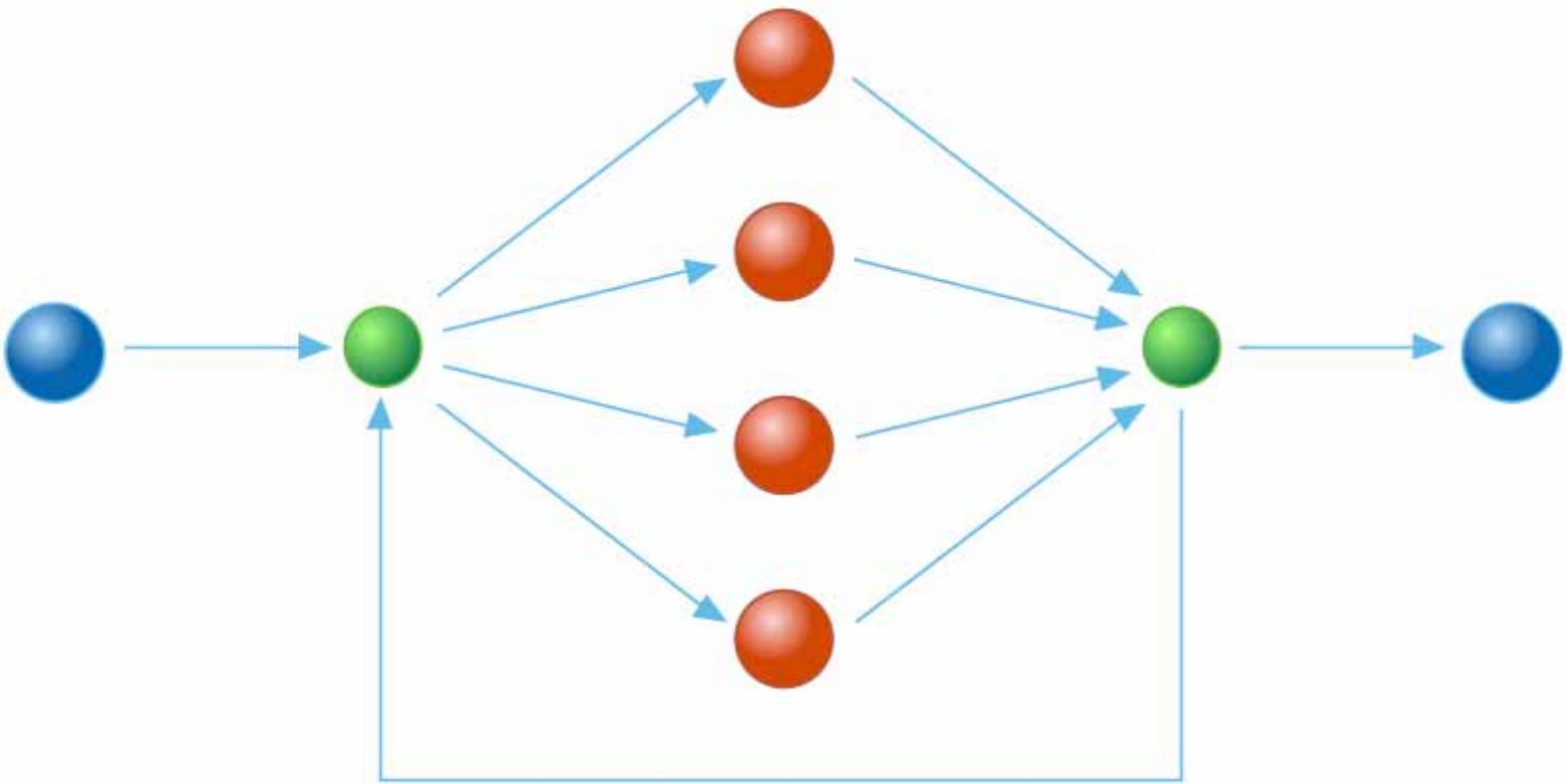
$$A(G)=\begin{pmatrix} 0 & b & 0 & 0 & 0 \\ 0 & 0 & d & 0 & c \\ g & 0 & 0 & h & 0 \\ 0 & 0 & 0 & x & f \\ a & 0 & 0 & 0 & 0 \end{pmatrix}$$

複雑な状態遷移モデルから

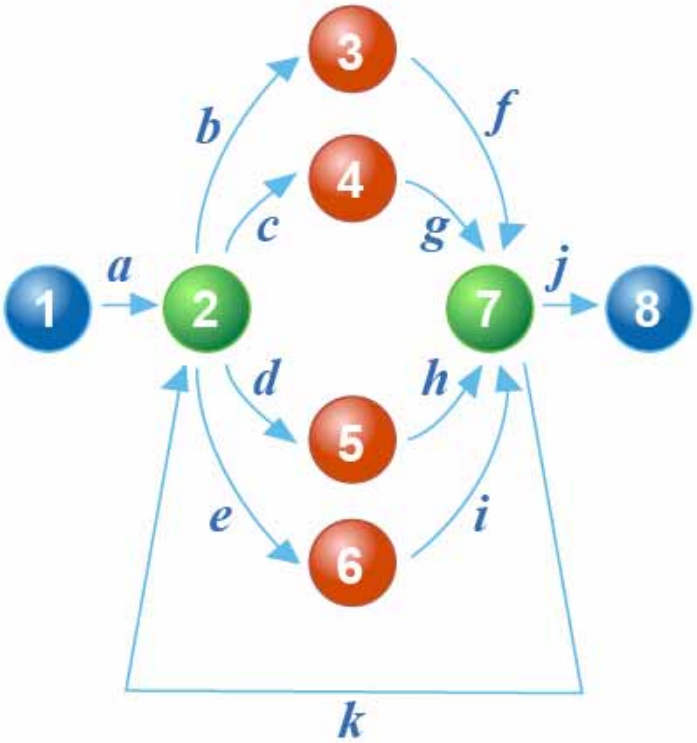
VALTES



シンプルな状態遷移モデルへ



状態遷移図を行列式へ変換する



$$\begin{pmatrix} 0 & a & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b & c & d & e & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & f & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & i & 0 \\ 0 & k & 0 & 0 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

テスト項目の因子・水準として抽出

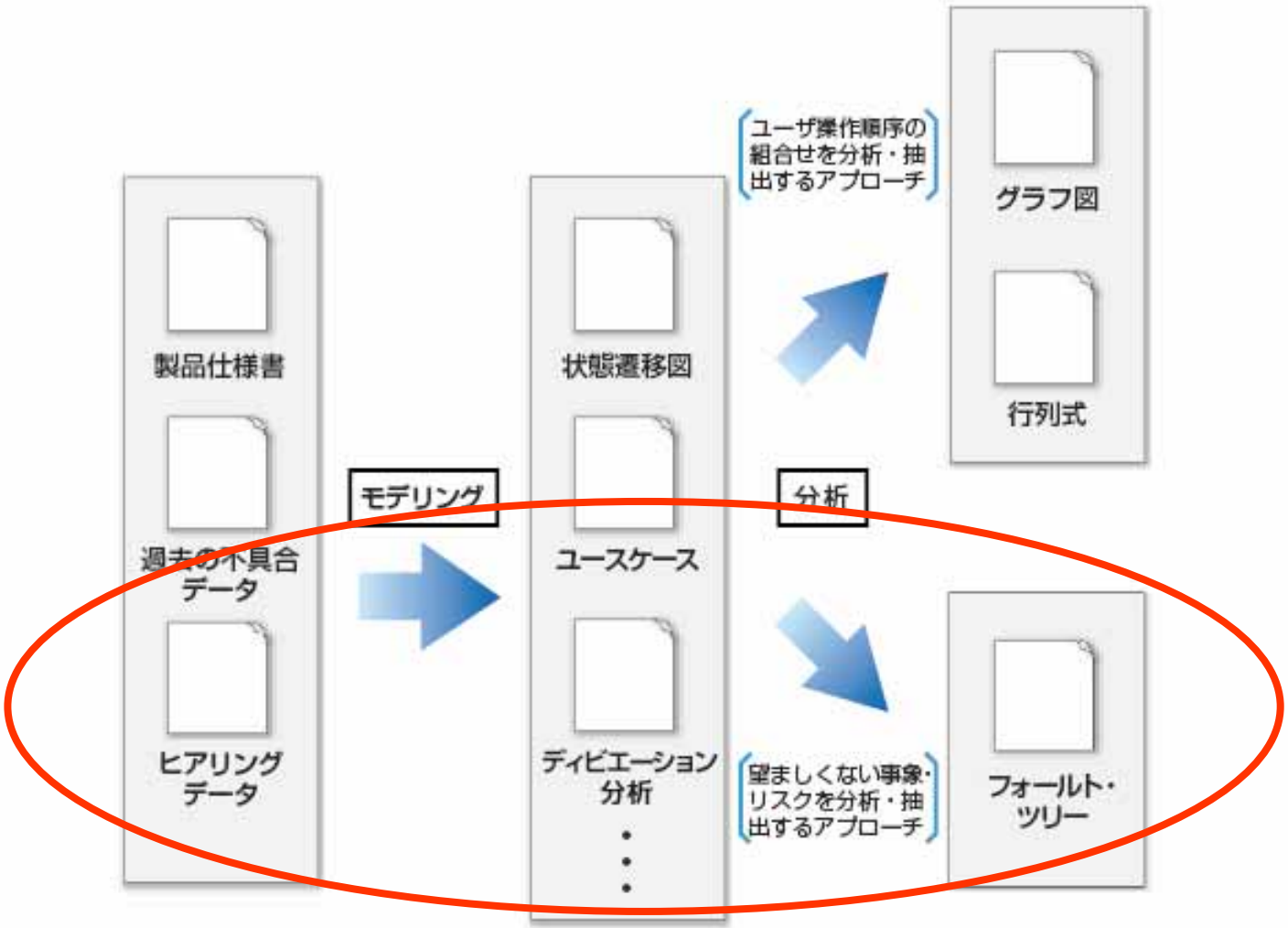
第4章
モデリング・アプローチ②
『回避すべきクリティカルな不具合をいかにしてテストで食い止めるか』

迫られる重大リスクの回避と信頼性の確保

- 相次ぐ重大不具合の発生と、問われる製造物責任
- 避けるべき致命的な不具合をいかに効果的に摘出するテストを設計するか

⇒**フォールト・ツリー分析**（FTA：Fault Tree Analysis）
による不具合原因の抽出

テスト設計作業の流れ

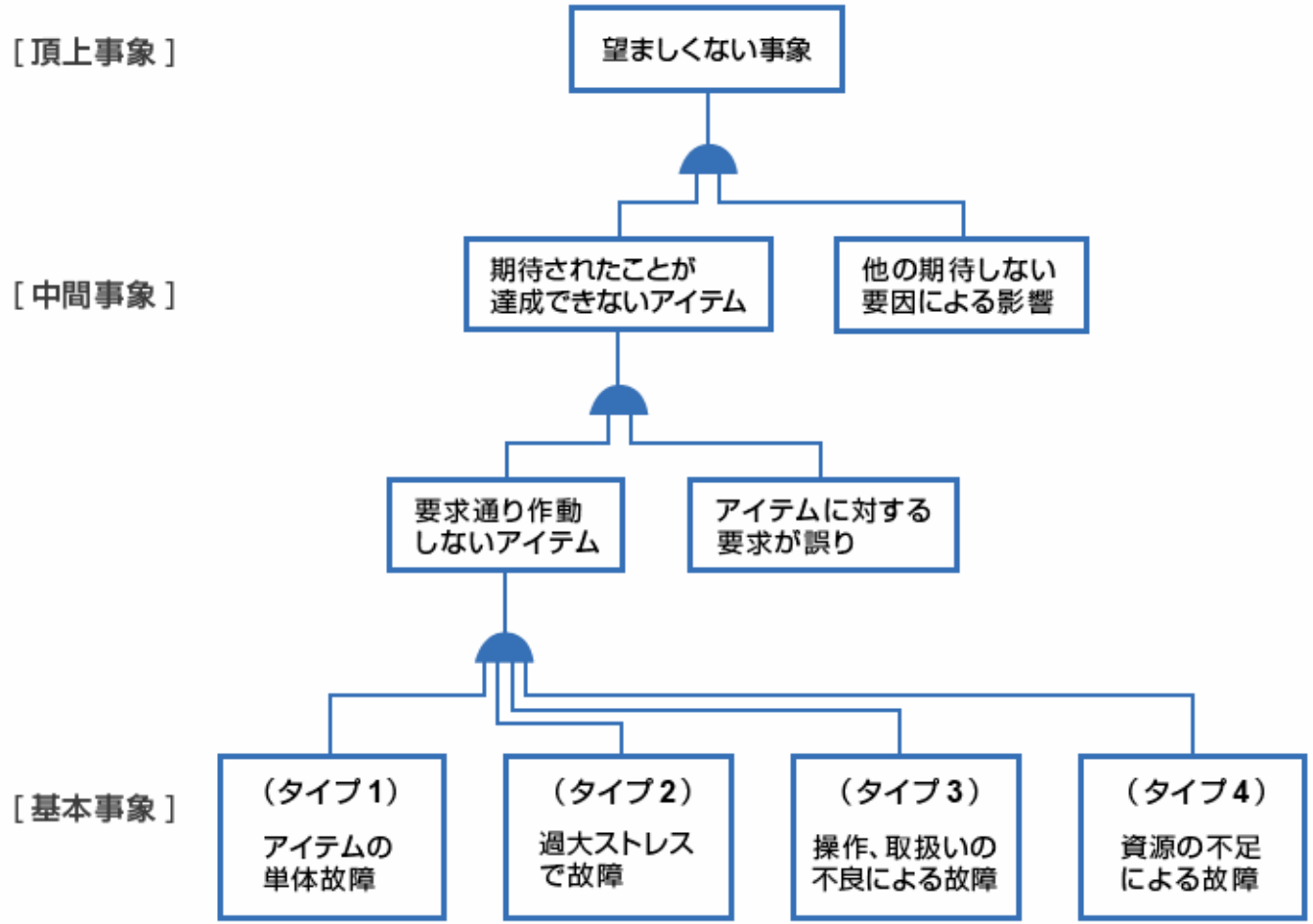


ガイドワードによる ディビエーション（逸脱）分析の例

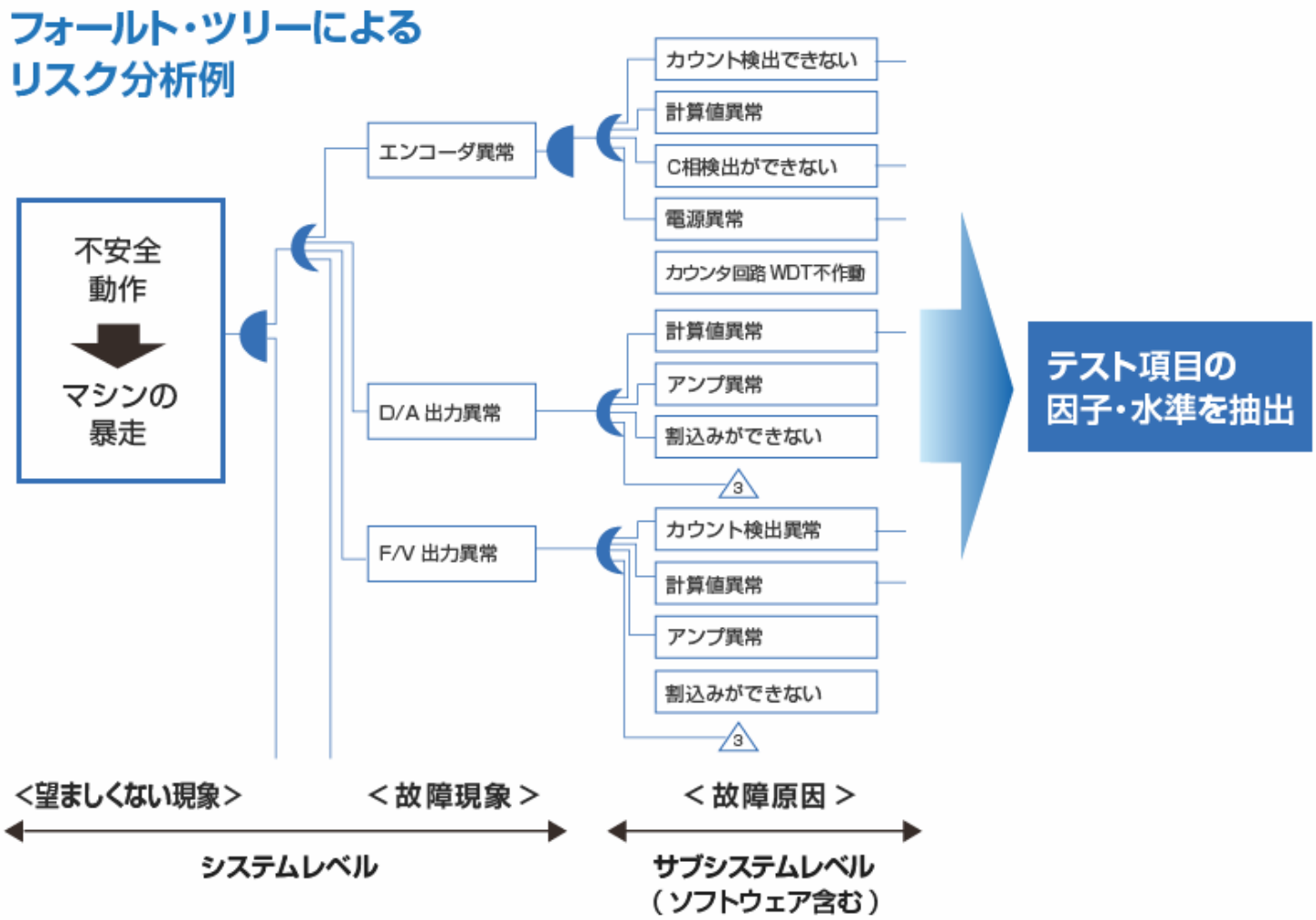
VALTES

逸脱の形	標準ガイド用語	電子システムの逸脱の例
否定	ない (No)	データがない
量的変化	より大きく (More)	計画より多い信号量
	より小さく (Less)	計画より少ない信号量
質的変化	同様に (As well as)	追加された／擬似信号
	ある一部に (Part of)	データ、制御信号が不完全
代わり	反対に (Reverse)	信号が逆方向に流れる
	異なる (Other than)	データ、制御信号の量は十分、しかし不正確
時間	早い (Early)	信号が早く到達
	遅い (Late)	信号が遅く到達
シーケンス	前に (Before)	信号がシーケンスより早く到達
	後に (After)	信号がシーケンスより遅く到達

望ましくない不具合の選定と故障原因への分解過程

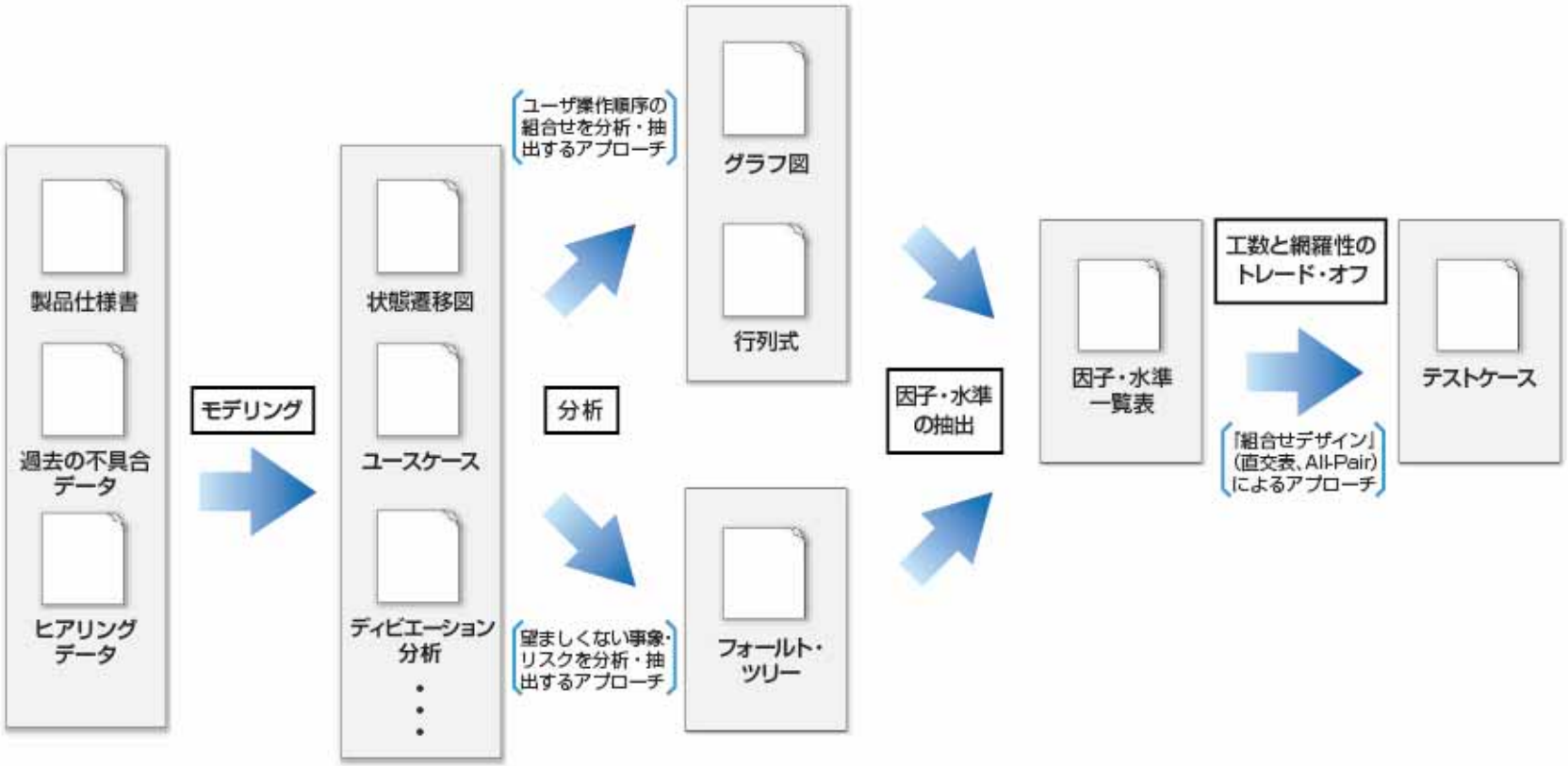


フォールト・ツリーによる分析



第5章
ソフトウェアテストの可能性の追求
～『バルテス標準』の深化と進化～

『バルテス標準』によるテスト設計作業の流れ



『バルテス標準』の特徴

『バルテス検証技術標準』は・・・

- 属人的なテスト技能を全技術社員の標準スキルへ
- さまざまなアプローチによる組合せテストにより、網羅性の向上と工数削減を柔軟に実現
- IEEE829に準拠した高い客観性のテストドキュメントを作成
- 再利用性、レビューのしやすさ、トレーサビリティを追求した開発資産としてのテストドキュメント
- 検証の観点より上流工程からの品質の作り込みをサポート

本日はありがとうございました