FeliCa Networks
&
Debug Engineering

All-pair法を応用した携帯電話組み込み用モバイルFeliCa IC チップファームウェアの評価に関する報告

フェリカネットワークス株式会社　太田 豊一、栗田 太郎

デバッグ工学研究所　松尾谷 徹

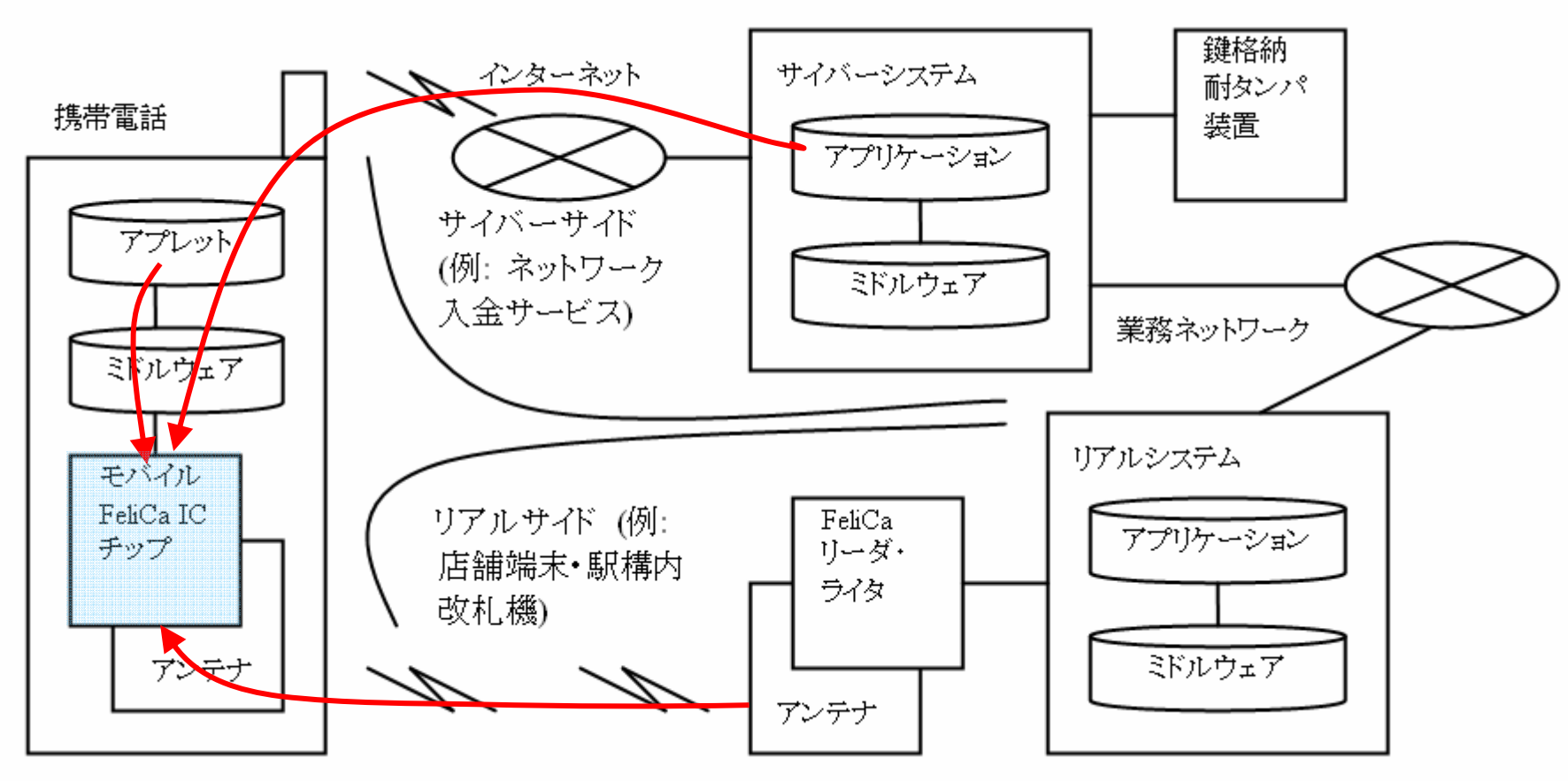
2006.05.11

■ プロジェクトのご紹介 （1 / 3）

- 携帯電話組込み用 モバイル FeliCa IC チップファームウェアの開発
  - “おサイフケータイ” を実現する、携帯電話搭載の IC チップのファームウェアの開発
  - 搭載 IC チップは、非接触 IC カード技術 “FeliCa” と互換を持つ
  - 電子現金、定期券、会員証、クレジットカードなど様々なサービスを実現できるセキュリティを備えたファームウェア
  - “かざす” という “リアルサイド” のサービス、“サイバーサイド” のサービスの融合

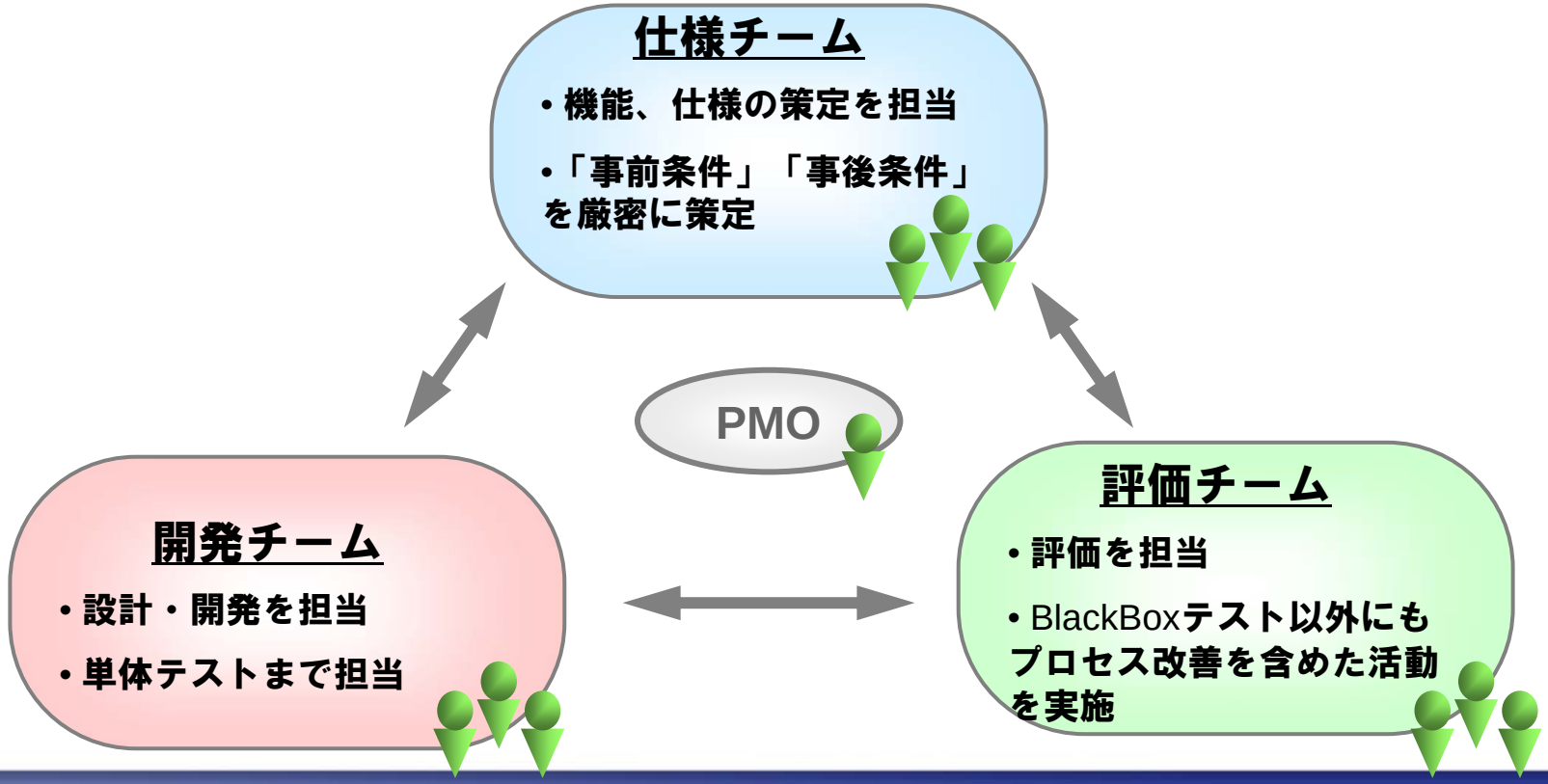
■ プロジェクトのご紹介 （2 / 3）

- システム構成図



■ プロジェクトのご紹介 （3 / 3）

- 開発体制
  - 仕様チーム･開発チーム･評価チームの 3 チーム体制で開発を推進
  - 各チームを支援する PMO を設置



■ ソフトウェア開発現場における多くの課題

1. 仕様に起因する課題
   - 「不明確な仕様」に起因するトラブル
   - 「上流工程での不良」修正コスト

   → 形式仕様記述言語の導入
   ソフトウェアシンポジウム2005 富山 発表
   ソフトウェアシンポジウム2006 熊本 発表予定
2. 開発に起因する課題
   - アーキテクチャ、アルゴリズムの妥当性証明

   → モデル検証試行
3. 評価に起因する課題
   - 機能増加に伴う「評価項目」の爆発
   - 抜け・漏れのない「評価項目」の作成

   → 今回発表内容について
4. 開発体制に起因する課題
   - 「コミュニケーション不足」に起因するトラブル

   → チームビルディングの実施

■ 評価に関する問題

- 機能増加に伴う「無限に近い組合せの評価項目」
- バグを効率的に見つける「評価手法の確立」
- 評価終了を見極める「品質指標の策定」
- 評価項目作成や評価実施の「人為的ミス」

  - 従来技術の限界 / 新技法適用の必要性

■ 評価合理化手法

- 評価項目の合理化手法
  - 「All-Pair法」
  - 「実験計画法」

  - 任意の 2 つの因子において、その水準がすべて均等に組み合わせられるように因子間の組合せを作成する手法

  - バグの多くは、少数パラメータ間の組合せによって顕在化する
    - R.D.Kuhn et al. “Software Fault Interactions and Implications for Software Testing” IEEE Transactions on Software Engineerring, 30(6), June
  - 200 万行の製品の場合、2 因子間バグ数 32 件 / 3 因子間バグ数 0.128 件 となる
    - 直交表を活用したソフトウェアテストの効率化, JaSST’05, 富士ゼロックス 秋山 浩一

All-pair 法とは？

- 評価すべきパラメータ (= 因子) に対して、2 つのパラメータの組合せを網羅したテストケースの作成方法

	因子			
	A	B	C	D
No.1	1	1	1	1
No.2	1	2	2	2
No.3	1	3	3	3
No.4	2	1	2	3
No.5	2	2	3	1
No.6	2	3	1	2
No.7	3	1	3	2
No.8	3	2	1	3
No.9	3	3	2	1

- テストケースの例
  - パラメータ (= 因子) 4 個
  - 値の種類 (= 水準) 3 状態
- テスト数
  - すべての組合せ 81 通り (3^4)
  - All-Pair 法 9 通り

■ All-pair 法によるテストケースの作成方法

◆ 4 因子 3 水準の例

水準 V = 3 , 組合せを i , j

1. i を P=3 に割り付ける
2. (i * P + j) % V に割り付ける

		因子			
i	j	P=3	P=2	P=1	P=0
1	1	1	1	1	1
1	2	1	2	2	2
1	3	1	3	3	3
2	1	2	1	2	3
2	2	2	2	1	1
2	3	2	3	1	2
3	1	3	1	3	2
3	2	3	2	1	3
3	3	3	3	2	1

■ All-pair 法のツール

◆ テストケース作成ツール

- *Jenny : a pairwise testing tool*
  - http://burtleburtle.net/bob/math/jenny.html
- Telcordia : AR Greenhouse
  - http://aetgweb.argreenhouse.com/
- SATISFICE : Test Tools
  - http://www.satisfice.com/tools.shtml

◆ 因子と水準を与えると、テストケースが自動生成される

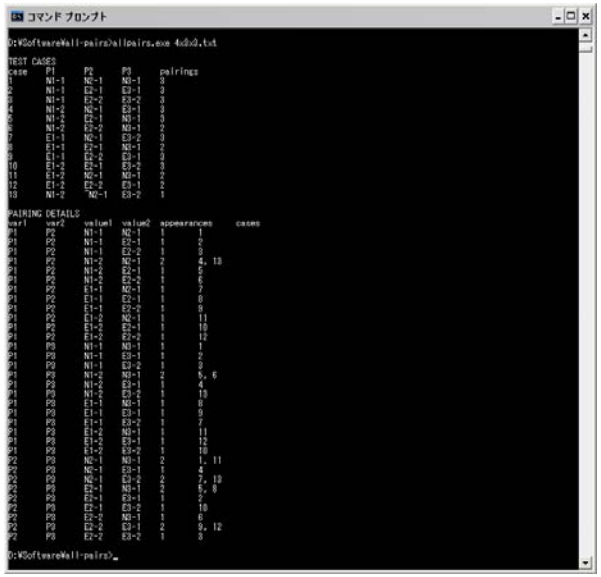
■ All-pair 法によるテストケースの作成方法 – ツール利用 –

◆ SATISFICE : Test Tools

– ALLPAIRS Test Case Generation Tool (Version 1.2.1)

入力データ

P1	P2	P3
N1-1	N2-1	N3-1
N1-2	E2-1	E3-1
E1-1	E2-2	E3-2
E1-2		



テストケース

TEST CASES

case	P1	P2	P3	pairings
1	N1-1	N2-1	N3-1	3
2	N1-1	E2-1	E3-1	3
3	N1-1	E2-2	E3-2	3
4	N1-2	N2-1	E3-1	3
5	N1-2	E2-1	N3-1	3
6	N1-2	E2-2	N3-1	2
7	E1-1	N2-1	E3-2	3
8	E1-1	E2-1	N3-1	2
9	E1-1	E2-2	E3-1	3
10	E1-2	E2-1	E3-2	3
11	E1-2	N2-1	N3-1	2
12	E1-2	E2-2	E3-1	2
13	N1-2	~N2-1	E3-2	1

■ 実験計画法とは？

L8 直交表

	因子						
	A	B	C	D	E	F	G
No.1	1	1	1	1	1	1	1
No.2	1	1	1	2	2	2	2
No.3	1	2	2	1	1	2	2
No.4	1	2	2	2	2	1	1
No.5	2	1	2	1	2	1	2
No.6	2	1	2	2	1	2	1
No.7	2	2	1	1	2	2	1
No.8	2	2	1	2	1	1	2

- 2 つのパラメータの組合せを網羅した組合せ表、直交表を利用
- 直交表は、因子の数に応じて L8 , L16 , L32 など用意されている
- 評価すべきパラメータを直交表の因子に割付けテストケースを作成する

■ 評価合理化手法の比較 – 評価対象への網羅性 –

実験. 評価対象の網羅性

- 特定モジュールに対するテスト項目を作成
- 作成されたテストケースにより評価を実施し、対象モジュールのカバレッジ (網羅率) を測定

- **比較対象**
  - 実験計画法によるテストケース作成
  - All-pair法によるテストケース作成
  - 従来手法によるテストケース作成
    - 仕様に精通した担当者が、仕様を評価するテストケースとして作成

■ 評価合理化手法の分析 – 評価対象への網羅性 –

- 分析結果
  - 「従来手法」に比べ「実験計画法」「All-pair法」によるテストは，カバレッジ率が優れていること，およびテスト項目数が半分程度と効率的であること
- 考察
  - 従来手法は、しっかりとした評価を行わなければとの心理的要素により多くの組合せを選んでいる
  - 不具合が発生しそうな組合せを重点的に実施し、発生しなそうな箇所の評価が抜けてしまう

		従来手法	実験計画法	All-pair法
テスト項目数		75 項目	34 項目	30 項目
カバレッジ	A 関数	100 %	100 %	100 %
	B 関数	91 %	91 %	91 %
	C 関数	100 %	100 %	100 %
	D 関数	90 %	100 %	100 %
	E 関数	73 %	73 %	73 %
	F 関数	76 %	91 %	91 %
	G 関数	87 %	89 %	89 %
	H 関数	96 %	98 %	98 %
	I 関数	96 %	96 %	96 %

■ 評価合理化手法の選定

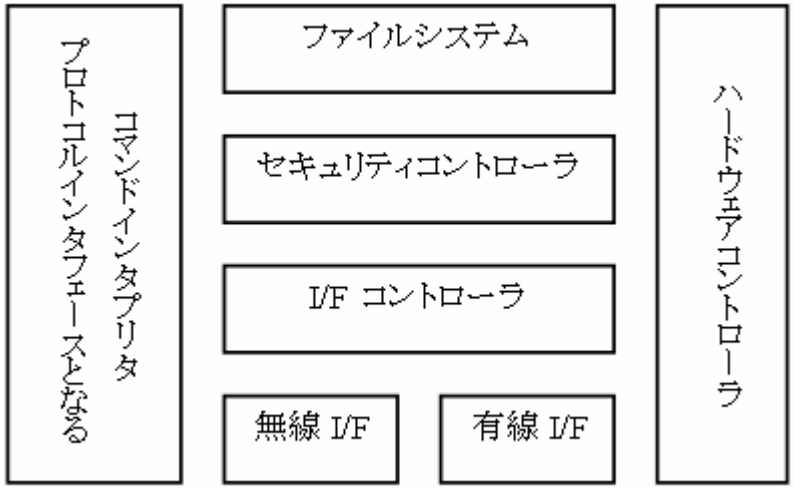
- まとめ
  - 評価合理化手法の効果は高い
    - テストケースの作成工数削減
    - テストケースの一定品質の確保
  - 「実験計画法」「All-pair法」の効果に顕著な差は見られない

- 合理化手法の選定
  - 今回の評価対象は、様々なセキュリティ機能・用途に応じたデータアクセス方法を持ち 因子数 / 水準数 が多い

    - 水準の多水準化が簡易な「 All-pair 法 」を選定

    - 評価対象の特徴に合わせた工夫が必要

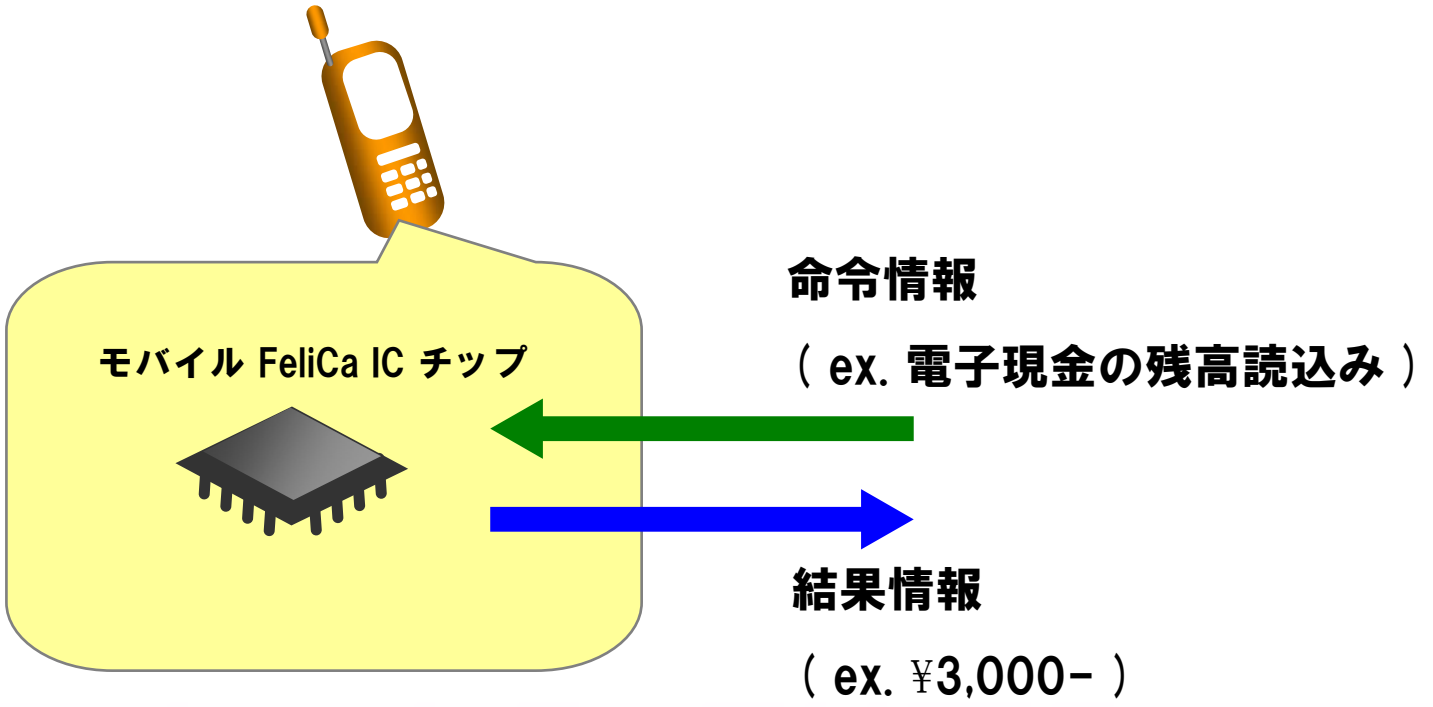
■ モバイル FeliCa IC チップファームウェアの構成



- **Windows のようなデータ管理**
  - ファイルシステム
- **SSL 通信のような認証管理**
  - セキュリティコントローラ
- **無線通信と有線通信の 2 つのインターフェース管理**
  - 無線 I/F , 有線 I/F , インターフェースコントローラ
- **ハードウェア制御を必要としたソフトウェア**
  - ハードウェアコントローラ
- **メモリデータへのアクセス制御・管理**
  - プロトコルインターフェースとなるコマンドインタプリタ

■ モバイル FeliCa IC チップファームウェアの特徴

- 基本動作
  - 通信インターフェースを通じて命令情報を受信・処理して、結果情報を返信する構成
  - 命令情報に含まれる引数と内部の状態をパラメータとして動作する



■ All-pair 法適用の課題

- 課題点
  - 引数には、正常に処理される有効な範囲のデータと、処理を中断する無効な範囲のデータが存在する
  - 無効な範囲のデータは、無効なデータを確認した時点で処理を中断し、エラー応答を返却する
  - 無効な範囲のデータも動作しないことを確認する必要がある

◆無効な範囲のデータを含めて「 All-pair 法 」を適用して評価項目を作成すると正しい評価が行えないテスト項目となる

■ All-pair 法適用の課題

サンプルパラメータ

P_1	P_2	P_3
有効1-1	有効2-1	有効3-1
有効1-2	無効2-1	無効3-1
無効1-1	無効2-2	無効3-2
無効1-2		

- 無効1-1 は、他の無効範囲のデータと重複するため、正しく評価が行えない

All-pair 法によるテストケース

No.	P_1	P_2	P_3
1	有効1-1	有効2-1	有効3-1
2	有効1-1	無効2-1	無効3-1
3	有効1-1	無効2-2	無効3-2
4	有効1-2	有効2-1	無効3-1
5	有効1-2	無効2-1	有効3-1
6	有効1-2	無効2-2	有効3-1
7	無効1-1	有効2-1	無効3-2
8	無効1-1	無効2-1	有効3-1
9	無効1-1	無効2-2	無効3-1
10	無効1-2	無効2-1	無効3-2
11	無効1-2	有効2-1	有効3-1
12	無効1-2	無効2-2	無効3-1
13	有効1-2	有効2-1	無効3-2

■ All-pair 法適用の課題解決

- 課題の解決
  - 課題解決のため禁則回避が必要
  - 組合せの禁則回避として、結合が知られているが、すべての因子を禁則回避することは困難なため、別の禁則回避を検討する必要がある

  • 禁則回避の工夫が必要

■ 課題解決

- **無効な範囲データの特性**
  - 入力パラメータに１つでも無効データが存在した場合，処理を中断する
  - 複数の無効データが存在した場合でも，１つの無効データを検知した時点で，続くデータを確認せず処理を中断する

    - 無効な範囲データのテスト項目は，有効な範囲データのテスト項目に1つずつ無効データを割り付ける

■ 禁則回避方法

サンプルパラメータ

P_1	P_2	P_3
有効1-1	有効2-1	有効3-1
有効1-2	有効2-2	有効3-2
有効1-3	無効2-1	無効3-1
無効1-1	無効2-2	無効3-2
無効1-2		

有効系のみでテストケース生成

All-pair 法によるテストケース

No.	P_1	P_2	P_3
1	有効1-1	有効2-1	有効3-1
2	有効1-1	有効2-2	有効3-2
3	有効1-2	有効2-1	有効3-2
4	有効1-2	有効2-2	有効3-1
5	有効1-3	有効2-1	有効3-1
6	有効1-3	有効2-2	有効3-2

無効系を割付

No.	P_1	P_2	P_3
1	無効1-1	有効2-1	有効3-1
2	無効1-2	有効2-2	有効3-2
3	有効1-2	無効2-1	有効3-2
4	有効1-2	無効2-2	有効3-1
5	有効1-3	有効2-1	無効3-1
6	有効1-3	有効2-2	無効3-2

- 評価項目数　12 項目

■ All-pair 法導入の効果

- 前世代のファームウェア評価と比較し、以下の結果が得られた

	前世代ファームウェア	適用ファームウェア
機能比率	1.00	2.37
テスト項目比率	1.00	0.52

- 評価項目の効率化を実現
- テスト項目の自動作成による抽出時間の短縮
- 評価項目作成の抜け漏れがなくなり、テストの信頼性向上

■ 課題

- 評価対象の特性に合わせた評価が必要
  - 機能面の評価に適用したが、セキュリティ面の評価、タイミング面の評価では、ある程度の組合せを実施する必要があった。

- 仕様変更による因子の追加時のテストケース整備が難しい

■ まとめ

- 「All-pair 法」を導入することで評価項目数を削減できた
- 「All-pair 法」により網羅的なテストケースが作成できた
- 実際のテストケースを考慮し、「評価されない項目」が生じないような工夫を行った
- 評価対象の特性に応じた工夫や他の評価手法との併用が必要