

実務家のための形式手法

厳密な仕様記述を志すための形式手法入門 第二版

事例：実証実験

形式手法導入に関する実証実験から、自らの形式手法導入の検討や計画立案の際に参考となる知見を得る。

■ 事前知識・経験

- 形式手法の導入を検討、計画している
- 形式手法の有用性についての基礎知識
- 形式手法導入のガイダンス

■ 学習目標

- 実証実験から、自らの形式手法導入の検討や計画立案の際に参考となる知見を得る

■ 主な学習項目

- 実証実験の概要と結論
- 形式手法で欠陥が見つかるパターン例

具体的事例：形式手法適用実証WG実証実験事例

- IPA/SEC 「形式手法適用実証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 (実験者)
	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(設計書提供者)	設計書の提供 指摘事項の評価 実験結果の評価
	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	

DSFのメンバ
が中心

※DSF(Dependable Software Forum)：

障害を起こさないソフトウェアの設計技術確立し、開発現場に普及させることを目的とした任意団体

■ 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。

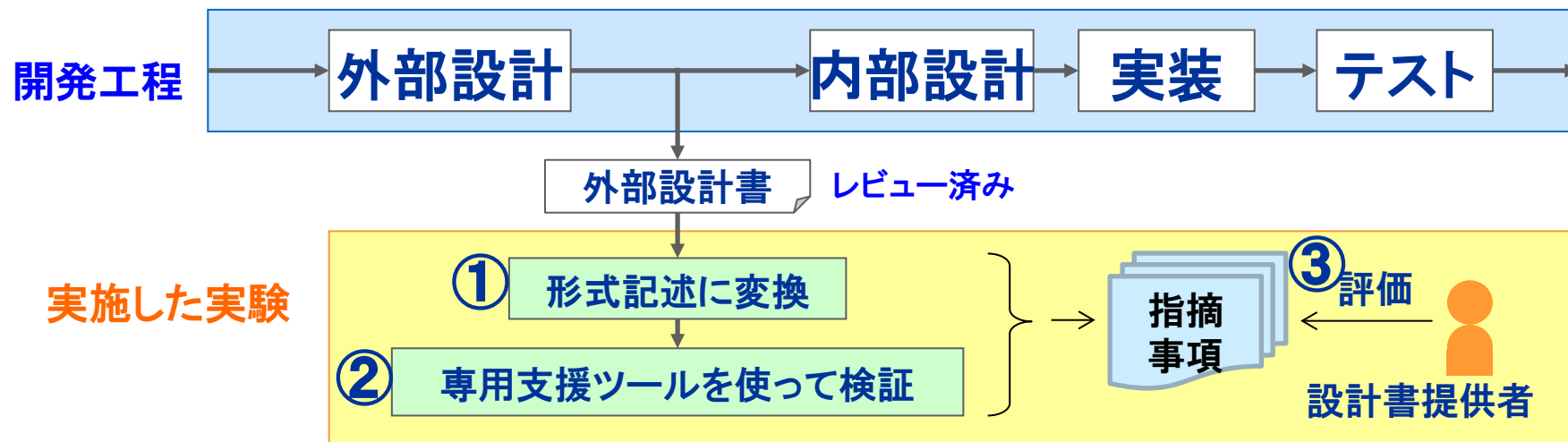
チーム	実験対象設計規模(ページ数)		実施体制 (人数)
	形式記述作成対象ページ数	参照を含む総ページ数	
Event-B (1)	110	707	1
Event-B (2)	106	287	3
Event-B (3)	49	381	1
SPIN	109	429	2
VDM++	300	700	5

■ 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

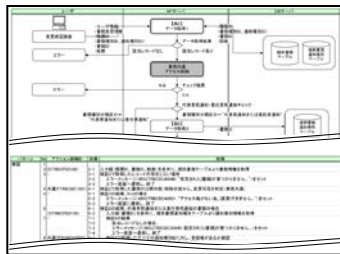


■ 形式手法を使う手順

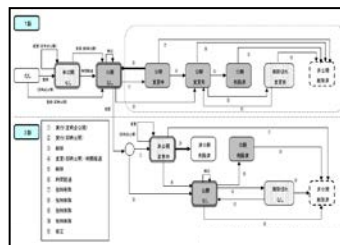
- 『形式手法活用ガイド』（DSF作成）に示される手順で実験を実施

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

- 設計書記述の不足、曖昧性
- 設計書間の不整合

を検査

形式仕様言語
で記述

```

: if
状態「なし」の場合
/* なし */ : atomic{
  (OmniDualPat == Pini) && (OmniDualDet == Dntg) ->
  if /* なし -> 未公開-なし (登録(定時)) */
  : event == Erga -> OmniDualPat = Puni
  /* なし -> 公開-なし (登録(即時)) */
  : event == Ergn -> OmniDualPat = Pst
  fi
}

状態「未公開-なし」の場合
/* 未公開-なし */ : atomic{
  (OmniDualPat == Puni) && (OmniDualDet == Dntg) ->
  if /* 未公開-なし -> 公開-なし (変更(即時公開)) */
  : event == Echm -> OmniDualPat = Pst
  /* 未公開-なし -> 非公開-削除済 (削除(強制削除)) */
  : (event == Edt || event == Etd) -> OmniDualPat = Pnds;
  OmniDualDet = Ddt
  /* 未公開-なし -> 変更 (定時点公開) */
  : event ==
  Echa -> skip
  fi
}

```

形式記述

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

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)

「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.)

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)	84.0	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%のプロジェクトにはほぼ収まる。

- 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

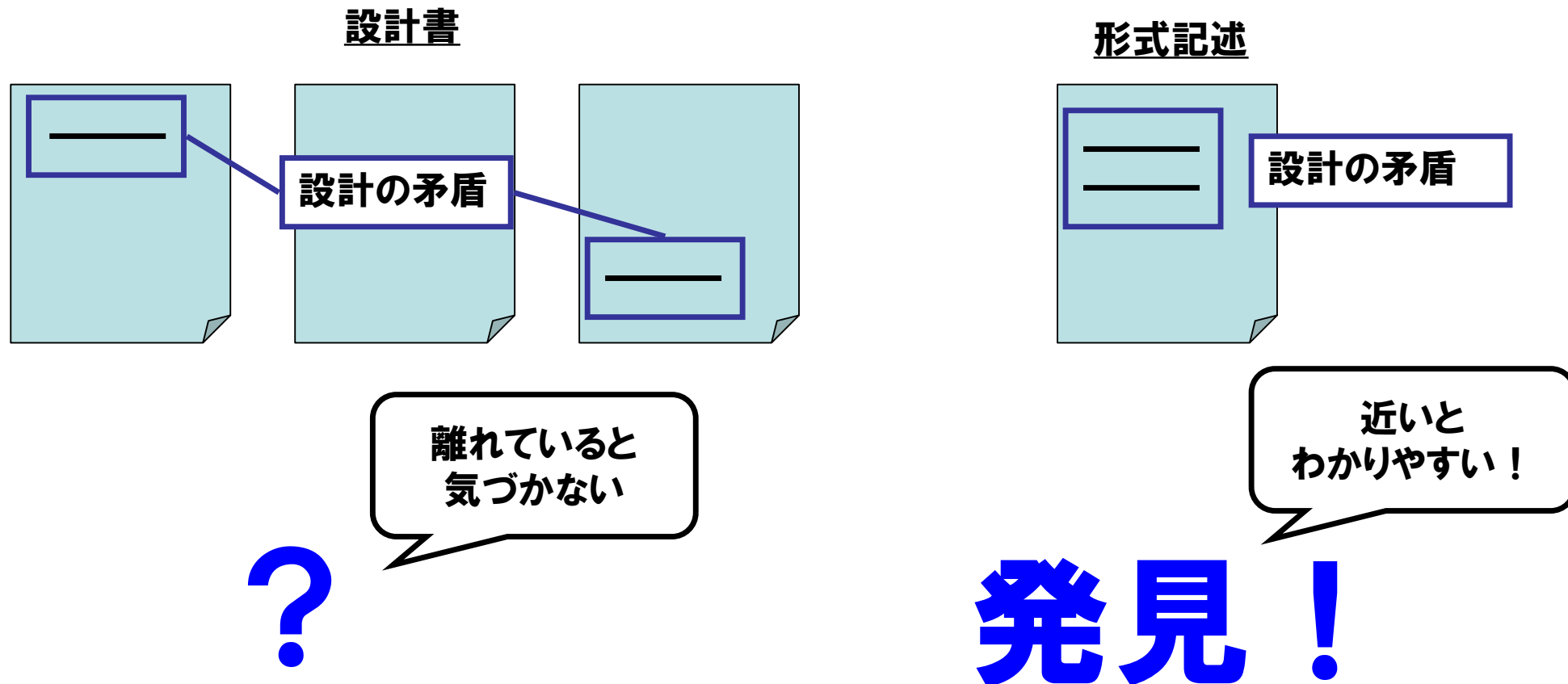
なぜ、形式手法で設計書の欠陥が発見できるのでしょうか？

■ 主要な3つの欠陥パターンを提示

1. 既存の設計書は「同じであるはずの仕様」が異なる種類の設計書に書かれている
2. 既存の設計書は暗黙知として存在する情報を確認することができない
3. 既存の設計書は自然言語で書かれるため、曖昧な表記で記述することができる

注意：設計書の欠陥発見はあくまでも形式手法適用効果の1つ

既存の設計書は「同じであるはずの仕様」が異なる種類の設計書に書かれている。



形式手法は異なる種類の設計書を1つの記述の中に書いて確認するため、異なる種類の設計書間で生じる欠陥を見つけることができる。

■「商品名」の説明を複数の設計書から抽出する

A設計書

■テーブル設計
「商品」テーブル

ID	1
テーブル名	商品

項番	属性名	論理データ型	
		データ型	桁数 文字数
1	商品名	文字列	10
2			
3			
4			

...

B設計書

■画面入力チェック
商品入力画面

ID	1
画面名	商品注文画面

項番	論理項目名	チェック内容	
		データ型	桁数 文字数
1	商品名	文字列	20
2			
3			
4			

...

C設計書

■ユーザの動作

ユーザは、「商品注文画面」で注文したい商品の商品名および数量を入力し、商品テーブルの”商品名”に格納する。

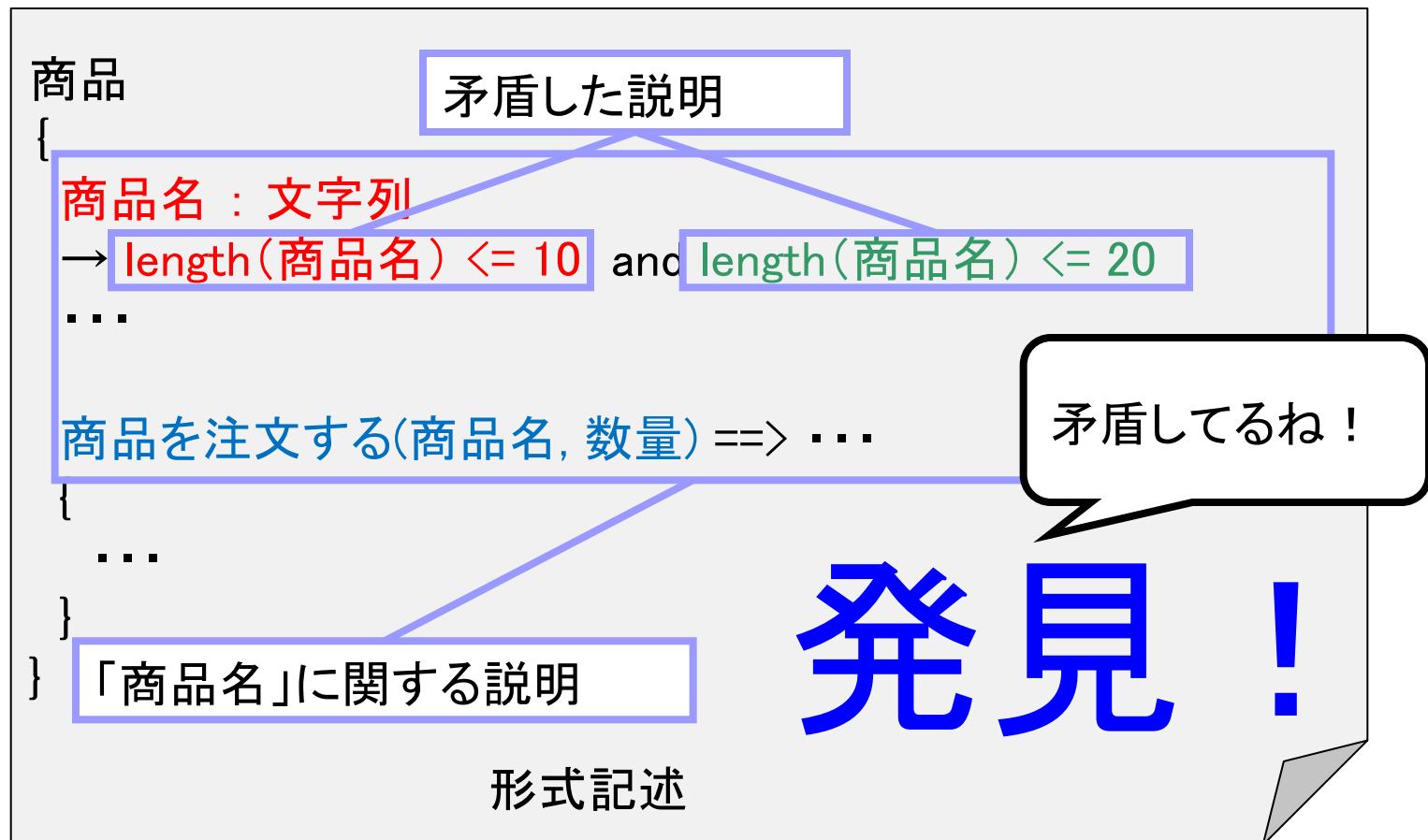
...

“商品名”の11文字以降の文字はテーブル上では削除されたり、エラーになる可能性有り

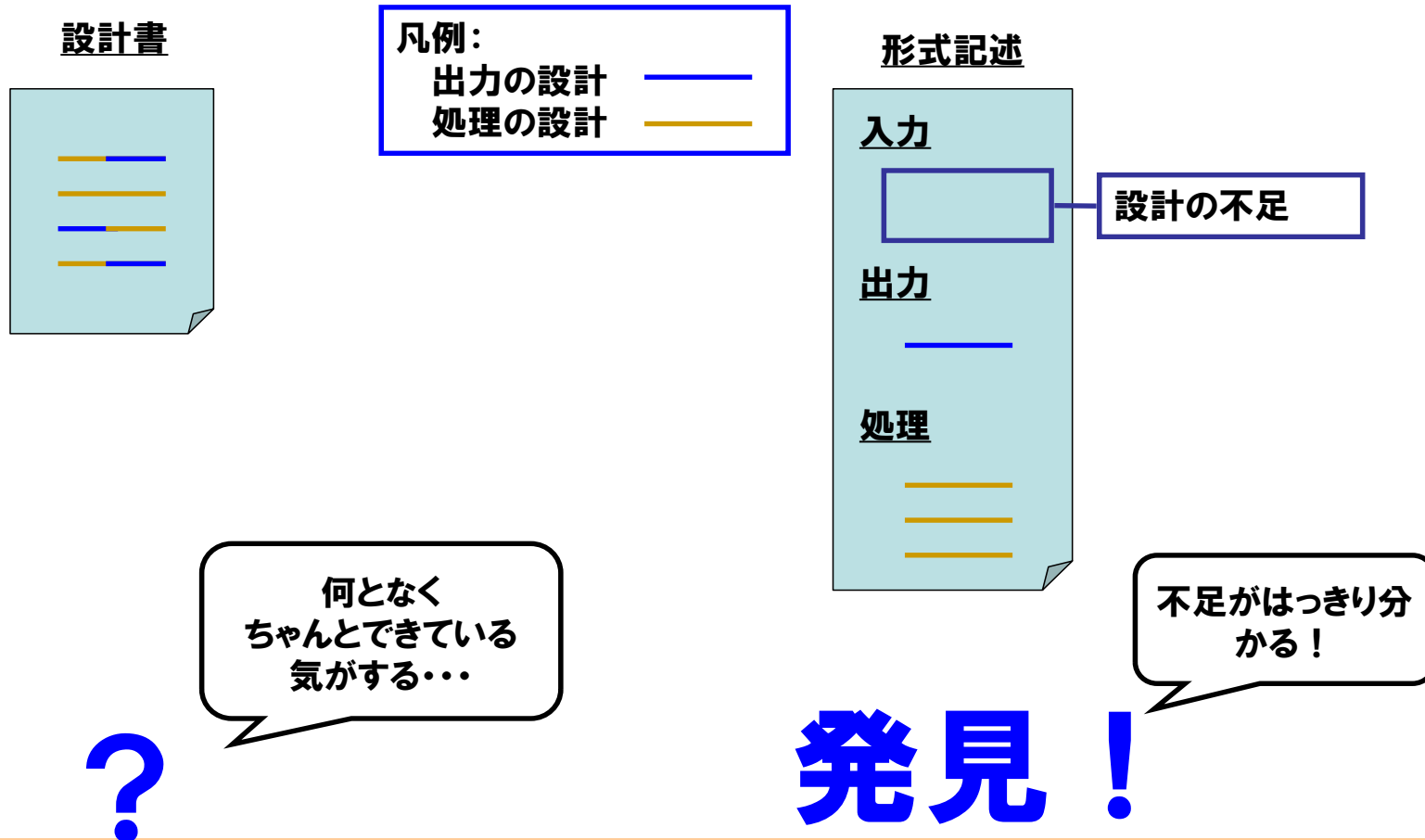
「商品名」に関する説明

出典:DSF資料

- 抽出した情報を形式記述で書く → 矛盾に気づく



既存の設計書は暗黙知として存在する情報を確認することができない



形式手法で記述したときは、暗黙知を明示的に書かなければいけない場合があるため、その場合に欠陥を見つけることができる。

■「商品名」の説明を複数の設計書から抽出する

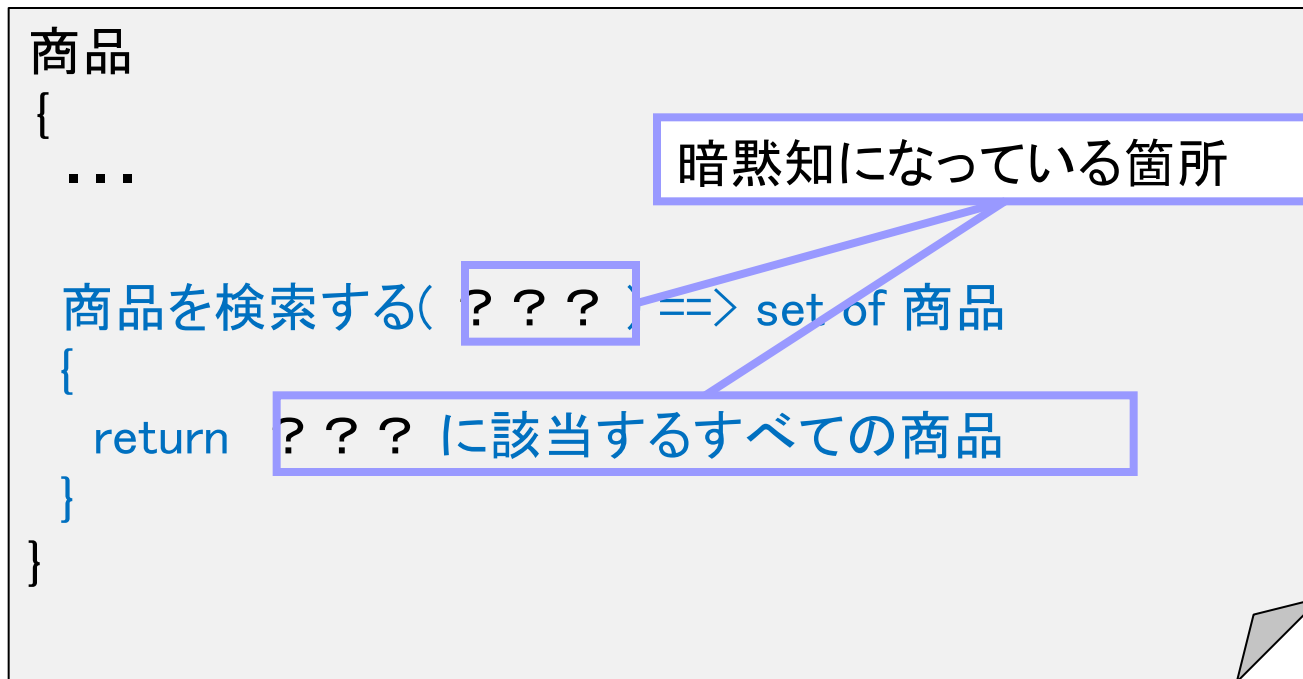
■ユーザおよびシステムの動作

1. ユーザは、「商品検索画面」で商品を検索する。
2. システムは、「商品検索画面」に商品の一覧を表示する。

「商品を検索する」の説明

設計書A

■ 抽出した情報を形式記述で書く

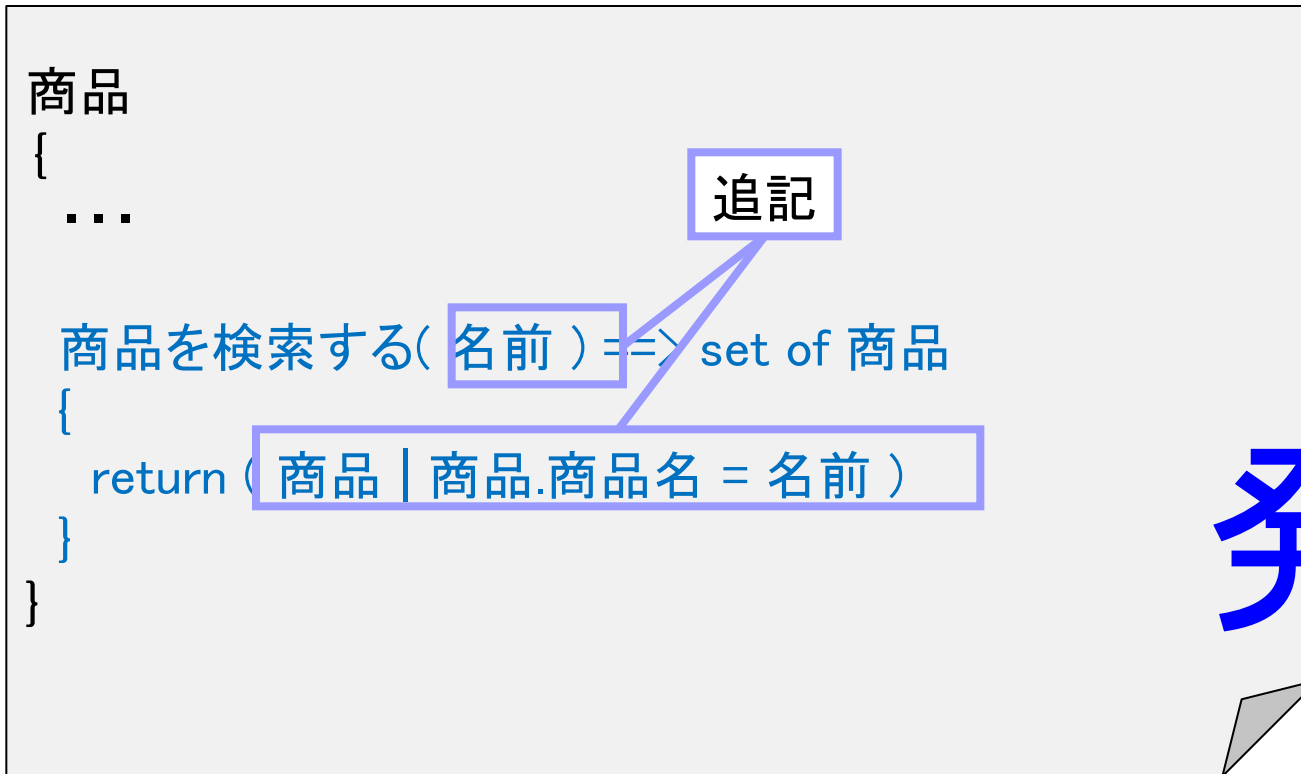


形式記述

何を検索条件とする
のだろう???
うまく書けないな...

?

■ 矛盾に気づく

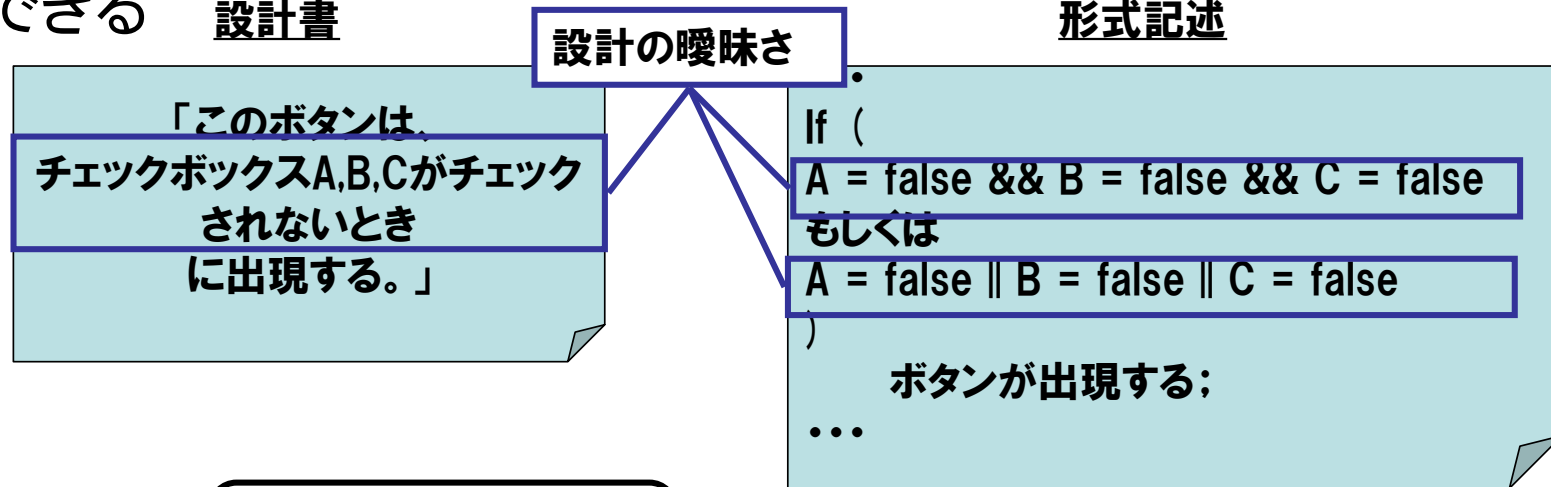


形式知に
できた！

発見！

形式記述

既存の設計書は自然言語で書かれるため、曖昧な表記で記述することができる



すっきりしないけど
何となく書けちゃった

2通りの解釈ができる。
設計が曖昧だ！

？

発見！

形式手法は厳密なルールによる数理式で記述されるため、曖昧な部分は記述することができず、欠陥として抽出される。

形式手法導入に関する実証実験から、自らの形式手法導入の検討や計画立案の際に参考となる知見を得るために、以下を学習

- 実証実験の概要と結論
- 形式手法で欠陥が見つかるパターン例

実務家のための形式手法

厳密な仕様記述を志すための形式手法入門

事例：実証実験

独立行政法人情報処理推進機構

技術本部 ソフトウェア・エンジニアリング・センター

統合系システム・ソフトウェア信頼性基盤整備推進委員会

上流品質技術部会 人材育成WG(編)

2013年3月 第二版発行

記載されている個々の情報に関する著作権及び商標はそれぞれの権利者に帰属するものです
なお、本書の内容は将来予告なしに変更することがあります