情報系の実稼働システムを対象とした
形式手法適用実験報告書

平成 24 年 4 月

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

形式手法技術部会
形式手法適用実証 WG

はじめに

システム・ソフトウェアが生活を支えるインフラとしての役割を増すなかで、その信頼性・安全性に対する要求が高まっている。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・センターの「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」では、高信頼システムの開発手法として形式手法の有効活用を促進するために「形式手法技術部会」を設置し、形式手法の導入実態調査や普及のための議論などを行ってきた。当部会の活動の一環として、開発現場が形式手法の導入を検討する際に参考とできる情報を収集することを目的として、2011 年 8 月に新たに「形式手法適用実証」WG を設置し、実在するシステムの設計書を対象として形式手法を適用する実験を行った。
本報告書は、形式手法適用実証 WG で行われた実証実験の報告と、実験結果に対して当 WG の議論で得られた知見をまとめたものである。本実験は 5 つの実験チームに分かれて行われた。各チームの詳細な実験報告は、別冊にまとめた。

情報系の実稼働システムを対象とした形式手法適用実験報告書

独立行政法人情報処理推進機構

目次

1. 実験の背景と目的

1.1 背景

ソフトウェアおよびそれらが構成するシステムの大規模化、高機能化は年々進み、あらゆる業界のビジネスを進める基盤として、また日常生活を支える社会インフラとして利用されるようになってきた。同時に、システムの不具合が経済や国民生活に与える影響も増大し、大きな社会問題となる事例も増えている。

このような背景から、システムに対する高信頼化の要求が高まっており、また、運用されるシステムの信頼性に関する説明責任も求められるようになってきている。

形式手法は、システムの高信頼化に有効な手法として長年注目されている。また、システムの信頼性をどのように確保したかについて、客観的証跡を与える手段としても注目されている。

しかしながら、国内では、とくにエンタープライズ系の開発では形式手法の実践的な活用のための方法論がまだ十分に確立されておらず、形式手法の活用が進んでいないのが現状である。形式手法が適切に活用されるようになるためには、「どのような課題に対して、どのように形式手法を適用すると、どのような効果があるか」ということを具体的に明らかにし、その情報を開発現場に展開して行くことが必要である。

1.2 目的

そこで、形式手法適用実証 WG では、エンタープライズ系のシステム開発に形式手法を利用した場合の工数や効果について開発現場が参考にできる情報を収集することを目的として、実際に開発され運用されているシステムを対象として形式手法を適用する実験を企画した。

形式手法の適用方法は様々なものがあるが、本実験は、リソースや期間等の制約を考慮し、形式手法を利用してソフトウェアを開発する実験ではなく、外部設計書の検査に形式手法を適用する実験とした。本実験で行う外部設計書の検査とは、後の工程において不正プログラムの原因や作業の遅延につながるような記述の曖昧性や不整合の有無を検査することである。本実験の大きな目的は次の二点である。

(1) 外部設計書の検査に形式手法を適用した場合にどのような効果があるかを具体的に明らかにする
(2) 外部設計書の検査に形式手法を適用した事例を作り、具体的な工数や手順を明らかにする

前者は、「なぜ形式手法を使うのか」という動機づけのための情報であり、後者は、「実際に使う場合はどうしたら良いか」についての情報である。いずれも、形式手法の適切な活用を促進するために必要な情報である。

外部設計書を適用対象としたのは、以下の理由からである。

- 従来の開発手法では、開発上流工程からの品質確保技術の確立が大きな課題となっており、外部設計書の品質向上が重要である。
- ユーザ側とベンダ側で確認・合意する最上流の文書であり、後工程での仕様変更や手戻りを防ぐために外部設計書の誤りや誤解を無くす方法の確立が重要である。
- 外部設計書の書き方に関するガイド [1] が存在しており、外部設計書を対象とした形式手法の適用方法は、そのガイドを参照することにより理解し易いと考えられる。

2. WG 活動の概要

本実験は、形式手法適用実証 WG の活動として、実験方法や結果の考察について議論しながら進めた。本 WG は、様々な立場の視点から形式手法の有効性や課題などの議論を行うために、ユーザ企業の委員、ベンダ企業の委員、および学識経験者の委員で構成した。ベンダ企業の委員は、ディペンダブル・ソフトウェア・フォーラム（以下、DSF と略す）の会員を中心とした。DSF は、エンタープライズ系ソフトウェアの信頼性と安全性向上をテーマとして民間企業 6 社 1 機関が共同で研究活動を行う団体であり、ソフトウェア品質向上のための 有力な手段として形式手法に着目し、実践的な適用方法の検討を行っている。本 WG では、ベンダ企業の委員が、DSF で検討された知見を利用して形式手法を適用する実験を実施し、その経過や結果について WG で議論する形式で活動を進めた。

WG は、2011 年 8 月から 2012 年 3 月までの期間に、ほぼ月 1 回のペースで 9 回開催した。WG では以下のような内容で議論を行った。

- 第 1 回（2011 年 8 月）～ 第 2 回（2011 年 8 月）
  - 実験の目的と、実施する作業内容についての議論
- 第 3 回（2011 年 9 月）～ 第 6 回（2011 年 12 月）
  - 実験の実施経過の確認と、結果の評価方法についての議論
- 第 7 回（2012 年 2 月）～ 第 9 回（2012 年 3 月）
  - 実験結果の分析方法と、考察・結論についての議論

3. 実証実験の概要

3.1 実験対象とした形式手法の使い方

形式手法は 1970 年代からはじまる長い歴史のある技術分野であり、数多くの形式手法が提案されると共に、さまざまな使い方が試みられてきた。特定の使い方を想定して考案された手法がある一方で、使い方を限定しない汎用な手法があるなど、多様性の幅が広い。このことが技術者の間に混乱を生じる原因ともなっている。たとえば、文献 [2]では、形式手法に関心のある人々へのアンケートの結果として、以下のような使い方が知られていると報告している。

・正しいシステムだけを系統的に開発
・記述に不具合がないことを数学的に証明し保証
・厳密な言語を用いることで仕様を明確化
・記述中に隠れている不具合を開発早期に発見

このように、形式手法への期待を様々な観点から述べることができる。一方で、実際に形式手法を用いる作業の観点からは、典型的な使い方を２つに整理して考えるとわかりやすい。すなわち、（１）構築あるいは「保証付き設計作業」、（２）検査あるいは「仕様書の検査作業」、である。以下、これら２つの使い方を説明する。

（１） 構築（Construction）あるいは「保証付き設計作業」

形式仕様言語を用いたデザイン記述を得ることを目的とする。当該形式手法が提供する形式検証（解析）の方法を利用して与えた性質が成り立つことを確認する。多くの場合、対象のラフスケッチや自然言語で書かれたインフォーマルな記述を出発点として、形式仕様を得ること、ならびに、性質を確認することの過程を通して、意図通りの正しさを持つ形式仕様記述を得る。最終的に得たデザイン記述は、形式仕様言語が定める正しさの基準ならびに与えた性質を満たすという観点で正しいことを保証できる。「保証つき設計作業」といえる。通常、表現力の大きな（汎用的な）形式仕様言語を１つ選ぶ。何を使うかという最初の決定が大切になる。特に、適用対象のソフトウェア開発工程を明確化することが重要である。

（２） 検査（Inspection）あるいは「仕様書の検査作業」

与えられた仕様書記載の内容を確認することを目的とする。仕様書からシステムの形式記述を抽出する「形式化の作業」、当該仕様書あるいは別途与えられた性質記述をシステム記述が満たすか否かを検査する「解析の作業」の 2 段階からなる。発見した不具合は、もとの仕様書の改訂に反映される。「検査済み仕様書」を得る作業と

いえる。経験的には、形式化の作業中に、もとになった仕様書の不備（曖昧さ、不整合、等）に遭遇することが多い。仕様書作成者と異なる観点（形式仕様作成という観点）から仕様書を読み込むことの効果である。なお、実際の仕様書では、システム機能と満たすべき性質の切り分けが難しいことが多い。「形式手法を援用した検査」の作業を容易にするような仕様書構成が望ましい。この使い方は、仕様書あるいはデザイン記述の新たなレビュー手段として形式手法を利用するもので、（１）に比べると部分的な形式手法適用が可能である。また、確認したい側面に応じて、複数の形式手法を使い分けることもできる。逆に、仕様書の内容や検査したい性質に応じた技術的な面での工夫が必要となる。

また、「正しさの基準」についての配慮が重要である。一般に、形式手法を用いることで、「正しい」記述を得ると期待される。しかし、「正しさ」とは何であるかという問いに一般的に答えることは難しい。さらに、用いる形式手法によって、「取り扱いやすい正しさの観点」が異なる。これらを無視して、「形式手法を使ったので正しい」というようなことはあり得ない。

本実験は、外部設計書に書かれた仕様を形式仕様言語を用いた記述に変換すること（形式化）と、支援ツールを利用して検証すること（形式検証）によって、外部設計書の記述において改善が必要と思われる事を指摘する方式で実施した（図 3-1）。前述の検査（Inspection）の使い方である。

なお、本報告書では、外部設計書の形式化と形式検証によって指摘された事を「指摘事項」と呼び、指摘事項のうち、外部設計書の提供者が実際に改善が必要と判断したことを「改善事項」と呼ぶことにする。

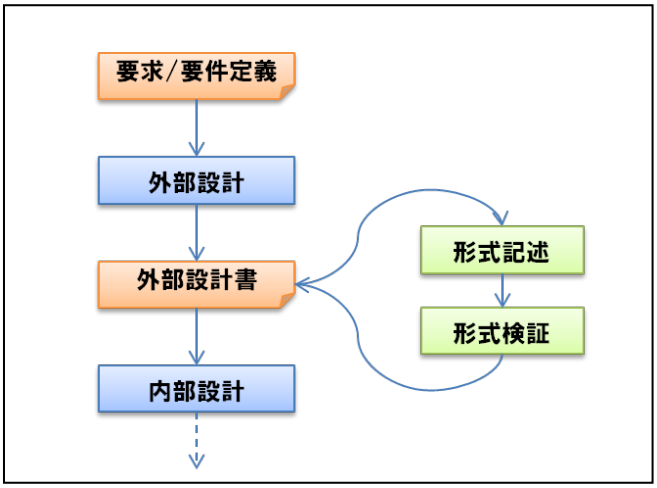

図 3-1　本実験における形式手法の適用方法

3.2 実験に使用した形式手法

開発現場での形式手法の適切な活用を促進することが本実験の目的であることから、実験で使用する形式手法は、利用実績が多く知名度が高いこと、解説書が入手しやすいこと、安定した支援ツールが入手しやすいこと、などの観点で以下の 3 種類の形式手法を選択した。

- Event-B
- SPIN
- VDM++

それぞれの形式手法の特徴を以下に示す。

✧ **Event-B**

Event-B は、B メソッド（Classical B）の後継として考案された形式仕様言語であり、形式仕様の構築ならびに検証を支援するツール RODIN が提供されている。Classical B が手続きとデータ構造の仕様記述を目的とするのに対して、開発上流工程でのシステム全体の機能振る舞いを対象とする。主として、要求モデリングに注力した「構築」作業で用いられることが多い。記述に集合や一階述語論理を用いており、リファインメントと呼ばれる考え方によって段階的に形式記述を具体化することが特徴である。

利用者は、対象システムを変数とそれらの値を変更するイベントの集まりとして記し、システムが満たすべき性質を不変条件として記す。自動証明あるいは対話的証明によって、システムが性質を満たすことを調べる。

Event-B は Classical B の考案者でもある Abrial 氏が考案し、欧州で研究開発が行われている。Web サイト(http://www.event-b.org/)から RODIN を入手できる。

✧ **SPIN**

SPIN は、システム振る舞いの時間的な変化を表す振る舞い仕様の表現と自動検証という特定の目的を達成する形式手法・ツールである。「構築」作業では、振る舞い検証に特化した補助的な手法として使われる。また、「検査」作業では、検査対象の仕様書から振る舞い仕様としてモデル化できる個所を選び出して用いる。

利用者は、相互作用し合う並行実行プロセスの集まりとして対象システムを記述し、「システムがいつでも特定の性質を満たす」あるいは「システムが動作する中でいつかは特定の性質が成立するようになる」といったシステムが満たすべき性質を線形時相論理(LTL)式として記す。 SPIN は、システムおよび性質の記述が与えられると、モデル検査法によって、システムの状態空間を網羅探索し、性質が満たされることを調べる。

SPIN は 1980 年代に研究が始まり、Holzmann 博士を中心に現在も開発がおこなわれている。Web サイト(http://www.spinroot.com/)から SPIN を入手できる。

✧ **VDM++**

VDM++は、オブジェクト指向概念を持つ VDM 形式仕様言語であって、仕様アニメーション(テスト実行)を主な検査法とする。プログラムのテスト同様の検査であり、産業界の技術者にとって理解しやすい。実行可能な設計仕様の確認が主な目的であることから、比較的詳細な仕様記述を対象とする「構築」作業で有用な成果が蓄積されている。利用者は、対象システムを変数や操作からなるクラスの集まりとして記し、システムが満たすべき性質をデータ不変条件や操作の事前・事後条件として記す。また、入力データや期待結果を記したテストスクリプトと合わせて、設計仕様を実行し、システムの振舞いが期待どおりであることを確認できる。 VDM は 1970 年代に IBM ウィーン研究所で開発された形式手法である。VDM++は産業界の利用に重点を置き、形式手法採用の困難さを軽減するため「形式手法の気軽な使い方（Formal Methods Light）」を指向している。Web サイト(http://www.vdmtools.jp/)から VDMTools を入手できる。

3.3 実験に使用した題材

本実証実験は、外部設計書に対して形式手法を適用する使い方で実験を行った。対象とする外部設計書は、株式会社東京証券取引所の協力により、実際に開発され運用されているシステムの外部設計書の提供を受けて利用した。実験で利用した設計書は、外部設計工程終了時の版であり、実際の開発においてプログラミング工程からテスト工程の間で施された修正は含まれていないものである。

3.3.1 実験に使用した主な設計書の種類

本実験では多くの種類の設計書を参照したが、設計書の内容を『機能要件の合意形成ガイド』 [1] で定義されている成果物に沿って整理すると、表 3-1 に示すとおりとなる。なお、#8 の「状態遷移図」は『機能要件の合意形成ガイド』 [1] で定義されていないが、3.3.2 で説明する「書類」の状態変化に関する設計書を指している。

表 3-1　実験に使用した設計書

#	実験に使用した設計書
1	画面一覧
2	画面遷移
3	画面レイアウト
4	画面入出力一覧
5	画面アクション明細
6	ER 図
7	エンティティ定義
8	状態遷移図

3.3.2　実験対象システムの概要

本節では、実験対象としたシステムの特徴について、今回の実験に関係する部分を要約して説明する。

実験対象のシステムは、「情報提供者」と「情報利用者」の間をネットワークで接続し、情報授受のサービスを提供する機能を持つ。授受される情報は、「書類」（文書ファイル）として「情報提供者」から提供される。「書類」は「掲載期限」や「公開種別」など様々な属性を持ち、その中には関連する他の書類へのリンクを示す「関連書類」という属性もある。「書類」に関する本システムの主な特徴を以下に示す。

①　「書類」に対する操作

本システムでは、「書類」に対して次のような操作を行うことができる。これらの操作は、基本的には「書類」の属性の値を更新する操作である。

・　登録、変更、修正、削除、強制削除

なお、本システムはこれらの操作以外に、登録された「書類」群に対して指定した条件で「検索」を行うことができる機能も提供する。

②　「書類」の状態遷移

「書類」は、①に示した操作により状態が遷移する。「書類」の状態とは、「書類」の属性のうち、「公開状態」、「削除状態」および「論理削除フラグ」の３つの属性の値によって定義される。

「公開状態」、「削除状態」、「論理削除フラグ」はそれぞれ以下のような値を取る。

属性	値
公開状態	未公開
	公開
	期限切れ
	非公開
削除状態	なし
	変更中
	変更有
	削除済
論理削除フラグ	OFF
	ON

図 3-2 「公開状態」、「削除状態」、「論理削除フラグ」の値

「書類」の状態遷移のうち基本的な部分を図 3-3 に示す。図 3-3 では、実線で示す四角形が１つの状態を表し、上段が「公開状態」、中段が「削除状態」、下段が「論理削除フラグ」の値を示している。状態間の矢印は、矢印に付記された操作により状態が遷移することを示している。

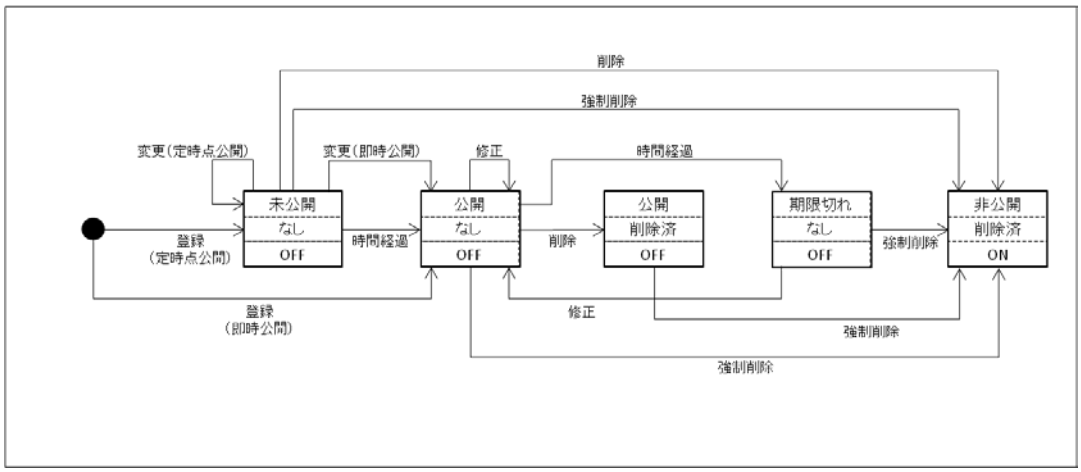

図 3-3 「書類」の基本的な状態遷移

③ 「書類」の公開方式

「書類」の公開方式には、「即時公開」と「定時点公開」がある。図 3-3 に示すように、「登録」や「変更」の操作において、「即時公開」の場合と「定時点公開」の場合とで状態の変化のしかたが異なる。「即時公開」の場合は、操作と同時に公開状態に変化する。他方、「定時点公開」の場合は、操作の時点では状態が変化せず、ある特定の時点まで時間経過した後に状態が変化する。

④　「書類」の改版

「書類」の種類によっては改版が行われる場合がある。公開状態/削除状態/論理削除フラグが公開/なし/OFF の状態で、「変更」の操作が行われた場合に「書類」の新しい版が作成される。図 3-4 は、第 N 版の公開/なし/OFF の状態で「変更（定時点公開）」または「変更（即時公開）」の操作が行われ、第 N+1 版が作成された場合の状態遷移を示している。第 N+1 版が作成された後、「書類」に対する操作は、最新版である第 N+1 版を対象にして行われる。ただし、第 N+1 版に対して行われる操作に応じて第 N 版も同時に状態遷移する。図 3-4 の破線で囲った部分は、第 N+1 版に対して行われる操作に同期して発生する第Ｎ版の状態遷移を示している。

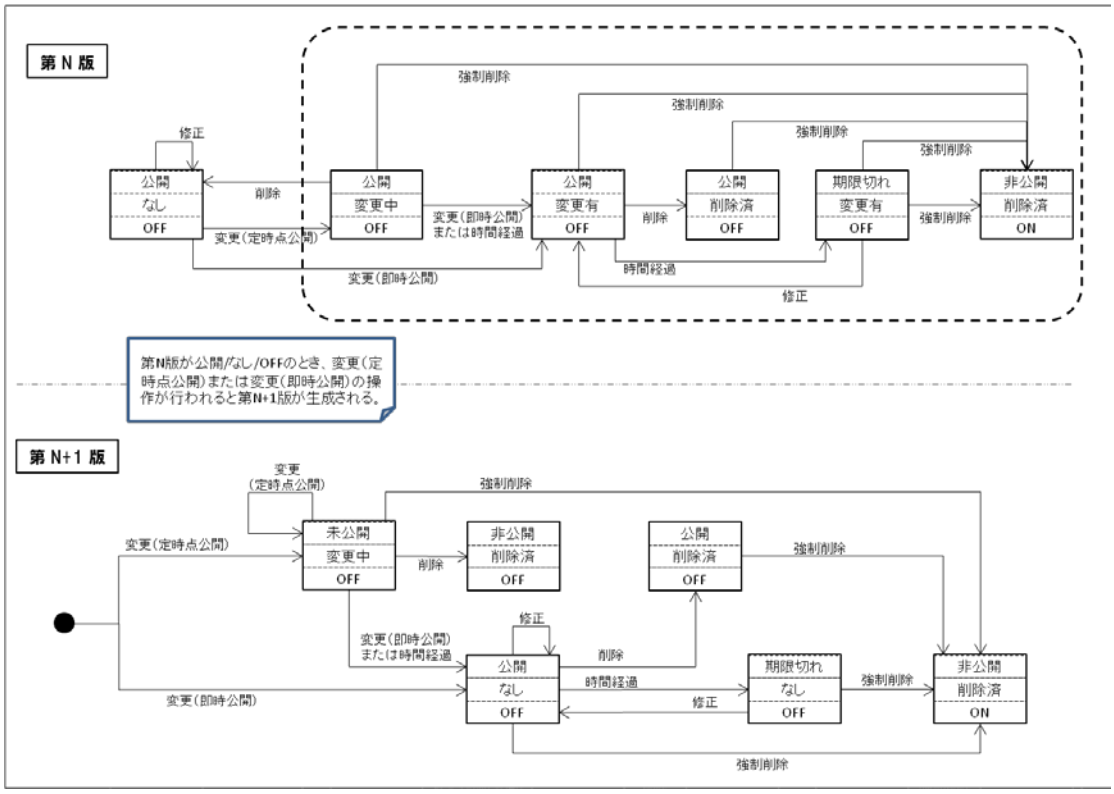

図 3-4　「書類」の改版が行われる場合の状態変化

4. 実証実験の仕様

本実験は、「1.2 目的」に示したとおり、形式手法の効果を明らかにすることと、具体的な工数を明らかにすることを目的とした。これらの 2 つの目的を満たすために実験の仕様をどのように設定したかを以下の節で説明する。

4.1 形式手法の効果の評価

前述のとおり、本実験は、外部設計書の検査に形式手法を適用する方法で実施した。具体的には、外部設計書の形式化と検証によって外部設計書の改善事項を発見することの試みである。したがって、本実験で実証しようとする形式手法の効果は「外部設計書の改善事項を発見できる効果」である。

外部設計書の改善事項を発見できる効果を実証するために、形式手法によって抽出された「改善が必要と思われること」（指摘事項）の評価方法を検討した。本 WG では、指摘事項を大きく 2 つの視点に分けて評価することにした。

(1) 形式手法適用者の視点による評価：

指摘事項の性質や特徴に関する分類を定義し、指摘事項をその分類に当てはめることによって評価を行った。形式手法自体の能力や性質、特徴を評価することを目的とした。

(2) 設計書提供者の視点による評価：

指摘事項の価値を判断する観点を定義し、評価を行った。実際の開発で発見されにくい問題を発見できるか、システム開発における影響度の大きい問題を発見できるか、などの観点の評価であり、実際の開発で形式手法がもたらす効果の一例を示すことを目的とした。

開発現場の直接の関心は(2)の評価にあると考えられる。しかしながら、実験によって明らかになるのは実験対象としたシステム開発における価値であり、あらゆるシステム開発にあてはまるような普遍的な情報にはならない。(1)の評価と合わせることにより、様々なシステム開発の現場で参考となる情報になると考えられる。

(1)の評価として、設計書の指摘事項の分類を表 4-1 に示すように定義した。

表 4-1　設計書の指摘事項の分類

指摘事項の種類	指摘内容
設計内容の衝突	設計書に書かれている内容に矛盾・不整合があること。 例： 『商品名は 10 文字以内である』という記述と『商品名は 10 文字以上である』という記述の両方が設計書の中に存在する。
設計内容の不足	本来設計書に書かれるべきことが書かれていないこと。 例： 『商品を検索する』と書かれているが、何をキーとして検索するかが書かれていない。
設計内容の曖昧さ	同一のものを複数の表現で記述する表記ゆれや、複数の解釈ができる文が存在すること。 例： 『このボタンは、チェックボックス A,B,C がチェックされないときに出現する』という文が設計書に存在し、次のいずれの意味かを判別できない。 ・チェックボックス A,B,C のいずれか１つ以上が「チェックされていない」状態を指している ・チェックボックス A,B,C の全てが「チェックされていない」状態を指している
誤字・脱字など	誤字・脱字などの表記に関する指摘。記述内容の意味に関する指摘ではない。

(2)の評価のために、指摘事項の価値を評価する観点を検討した。検討では次の点を考慮した。

［設計書の修正有無］

指摘事項に関して、実際の開発で外部設計以降の工程で設計書を修正済みであるかどうかを判定した。

［設計書修正の必要性］

指摘事項に関して、「同じシステムの開発を再度、新規に実施すると仮定した場合に、指摘について設計書を修正すべきか?」という観点で評価を行った。指摘事項が後の工程に与える影響度により、「修正が必要」、「修正が望ましい」、「修正不要」の 3 段階の評価を行った。

［発見時期］

指摘事項が、本システムの実際の開発で改善すべき問題として認識されていたか、認識されていなかったかの判定を行った。認識されていた場合は、システム運用を開始する前に認識されていたか、あるいは運用開始後に発見されたものであるかを判定した。これにより、実験で抽出された指摘事項が、形式手法を使わなかった場合にどの程度発見されにくいものであるかを評価することができる。

［暗黙の了解・ルールに関する指摘］

実験対象としたシステム開発では、「設計書には明記されていないが当事者は全員知っていた」という「暗黙の了解」や「暗黙のルール」と呼ぶべき事柄（以下、「暗黙の了解・ルール」と呼ぶ。）が存在した。本実験の実施者は暗黙の了解・ルールに関する情報を伝えられていなかったため、形式手法によって、暗黙の了解・ルールに関する事柄を指摘事項とする可能性があった。指摘事項がこれらの暗黙の了解・ルールに関するものである場合、実際の開発では問題として認識されていないため、発見時期の評価の対象にできない。そのため、暗黙の了解・ルールに関する指摘事項が明確となるように分類した。

なお、実験対象のシステムの開発時には暗黙の了解・ルールとされていた事柄も、開発文化の異なるオフショアに開発を発注した場合などを想定すると、設計書に明記すべきと考えられる場合もある。したがって、［設計書修正の必要性］の評価では、暗黙の了解・ルールに関する指摘事項も評価対象とした。

以上のような観点により、実験で抽出された指摘事項に対して、表 4-2 に示すような評価項目、評価基準で評価を行った。

表 4-2　設計書の指摘事項を評価する項目・基準

評価項目	選択肢	評価基準
設計書の修正有無	修正済	指摘事項について、実際に開発したプロジェクトの中では、設計書提供者が設計書を修正済みである
	未修正	指摘事項について、実際に開発したプロジェクトの中では、設計書提供者が設計書を修正していない
	修正対象外	改善事項に該当する指摘ではない
設計書修正の必要性	修正が必要	同じシステムを再度開発すると仮定した場合に、指摘事項について設計書を修正しないと不正プログラムが開発される可能性が高い
	修正が望ましい	同じシステムを再度開発すると仮定した場合に、指摘事項について設計書を修正しなくても不正プログラムが開発される可能性は低いが、設計書の理解容易性を高めるためには修正が望ましい
	修正不要	同じシステムを再度開発すると仮定した場合でも、指摘事項について設計書を修正する必要はない
発見時期	稼働前	指摘事項は、システム稼働前に発見されていた
	稼働後	指摘事項は、システム稼働後に発見されていた
	未知	今まで認識されていなかった指摘事項である
暗黙の了解・ルールに関する指摘	yes	設計書には明記されていないが、ユーザとベンダで合意が取れていた情報に関する指摘である
	no	上記に該当しない

4.2　形式手法の工数・手順に関する情報収集

本実験では、形式手法を適用する作業にどれだけの工数が掛かるかを記録することとした。作業名とその内容は、『形式手法活用適用手順（Event-B 編）【改訂版】』 [3]、『形式手法活用適用手順（SPIN 編）【改訂版】』 [4]、『形式手法活用適用手順（VDM++編）【改訂版】』 [5] の適用手順を参照いただきたい。形式手法を用いた作業は、大まかにまとめると、「文書読解と情報抽出」、「形式記述の作成」、「形式記述の検証」がある。検証による指摘事項に対して、形式記述の修正を行うなど実験の開始から終了まで、一連の作業を繰り返すことがある。その場合、1 回目の作業 A、2 回目の作業 A、…をまとめて作業 A の工数として報告する。

なお、単純に工数だけから、どの手法が生産性がよい、といったことは本実験では言えない。形式記述は構文・型チェックや検査条件（Event-B の証明課題、SPIN の LTL 式や assert 、

VDM++のテストスクリプトなど）に合ったものを作成する。形式記述の対象にした情報がリッチであれば作成工数は増え、検査条件の数が多ければ作成に加えて検証自体や指摘事項の原因分析に掛かる工数も増える。しかし、形式記述と検査条件の範囲でモレ無く検査できるので、品質は向上する。本実験の結果は、工数と形式記述および検査条件の量との相関を示すものではない。参考情報として見ていただきたい。

4.3 実証実験の実施体制

4.3.1 実験の参加者と役割

- 設計書提供者
  - 形式手法適用者による設計書の疑問点について回答する
  - 実験で得られた指摘事項について、重要性などの価値を評価する
- 形式手法適用者
  - 外部設計書を読解し、機能仕様の一部を形式化し、検証する
- 実験運営チーム
  - WG の議論をとりまとめ、実験の目的に沿った実験仕様を策定する
  - 実験が効率よく進められるように、設計書提供者と形式手法適用者との間の情報交換などの管理を行う
  - 実験結果を整理し、WG の議論をとりまとめた上で報告書を作成する

4.3.2 実験チーム

本実験では、設計書全体を実験対象とすることはリソースと活動期間の制約から難しいため、設計書の一部を検証対象として切り出して形式化・検証を行った。

また、形式記述の作成・検証作業を共同で行う単位として、実験チームを編成した。実験チームは、採用した 3 つの形式手法ごとに編成、さらに形式手法”Event-B”の場合は DSF『形式手法活用適用手順（Event-B 編）【改訂版】』 [3]に書かれている適用法も考慮して 3 チームを編成し、計 5 チームで実施した。詳細は以下となる。

表 4-3　実験チーム

チーム名	採用した形式手法	実験対象設計書規模		実施体制（人数）
		形式記述の作成対象ページ数	参照ページも含む総ページ数	
B1	Event-B	110	707	1
B2	Event-B	106	287	3
B3	Event-B	49	381	1
S	SPIN	109(重複分含めると 151)	429(重複分含めると 612)	2
V	VDM++	300	700	5

また、各実験チーム作業者の役割やスキルは表 4-4　のとおりであった。

表 4-4　実験チーム作業者の役割・スキル

チーム名	実施体制（人数）	作業者	役割*	スキル（経験）			
				業務 AP 開発	類似 AP 開発	形式手法	採用手法
B1	1	A	形式記述者 形式検証者	1 年	0 年	3 年	2.5 年
B2	3	A	形式記述者	15 年	0 年	3 年	3 年
		B	形式記述者	0 年	0 年	1 年	1 年
		C	形式検証者	5 年	0 年	7 年	1 年
B3	1	A	形式記述者	13 年	5 年	1.5 年	1.5 年
S	2	A	形式記述者	0 年	0 年	6 年	6 年
		B	形式記述者	2 年	0 年	5 年	0 年
V	5	A	形式記述者 形式検証者	2 年	0 年	5 年	5 年
		B	形式記述者 形式検証者	2 年	0 年	5 年	1 年
		C	形式記述者 形式検証者	6 年	6 年	20 年	2 年
		D	形式記述者 形式検証者	0 年	0 年	3 年	2 年
		E	形式記述者 形式検証者	37 年	25 年	17 年	12 年

4.3.3　実験の進め方

本実験は、チームごとに若干の差異はあるが、約 5 ヶ月の作業期間で以下のような流れで実施した。

ステップ	実施作業	作業概要	実施期間
1	実験対象範囲の検討	実験対象システムの設計内容把握のために、設計書を読解し、設計書提供者が提供した当時のレビューでの改善事項発見数を参考に実験対象範囲を確定する。	約 1 ヶ月
2	形式記述作成・検証	目的に合わせて形式記述作成と検証を行う。	約 3 ヶ月
3	指摘事項の評価	実験チームから抽出された指摘事項を設計書提供者が評価する。	約 1 ヶ月

下記に、各ステップの作業を詳細に説明する。

【ステップ 1】実験対象範囲の検討

まず、各実験チームが実験対象システムの設計書全体を読んで設計内容把握に努めた。また、設計書提供者が提供した実験対象システムのレビュー実績に対して、運営チームがシステム機能ごとにレビュー時の改善事項発見数をカウント、実験チームに提供した。実験チームは設計内容把握と機能ごとの改善事項数を考慮して、実験対象範囲を検討、確定した。

【ステップ 2】形式記述作成・検証

各実験チームは、実験目的に合わせて確定した実験対象範囲の形式記述作成、及び検証を実施した。各実験チームが抽出した指摘事項は表 4-1　設計書の指摘事項の分類　によって分類した。具体的な作業の手順は各実験チームの報告書(分冊)に記載されているので、参照されたい。

また、各実験チームによる設計書の疑問点を 2 週間に 1 回の頻度で運営チームが集計、設計書提供者に回答してもらうＱ＆Ａを実施した。Ｑ＆Ａは計 8 回実施し（ステップ 1，2 でも実施）、Ｑ＆Ａ総数は 91 件である。

【ステップ 3】指摘事項の評価

設計書提供者が各実験チームで抽出された指摘事項に対して、評価を実施した。評価項目は、表 4-2 の設計書の指摘事項を評価する項目・基準である。

また、本実験における情報共有は、以下の方針で実施した。

・実験の公平性を期すため、各チームで抽出された指摘事項は各実験チーム内での記録だけにとどめ、実験が終了した段階で他参加者にも公開する。

・実験チームが効率的に作業できるようにするため、Q＆Aの情報は実験チーム間や他参加者で共有する。

5. 実験結果

本章では、実施した実験について、得られた結果を示した上で考察を行う。5.1 節では実験結果の数値データを集計した結果を示す。それらのデータをもとに、5.2 節では形式手法の効果について考察し、5.3 節では形式手法を適用するための工数について考察する。

5.1 実験結果の集計データ

本節では、実験報告書(分冊)で取得した各実験チームの実験結果データを集計、整理した。整理したデーター覧は以下である。

・工数内訳[1]

・作業効率

・指摘事項抽出効率

・指摘事項の分類

・作業と指摘事項の関係

5.1.1 工数内訳

【B チームの工数内訳】

B1 チームから B3 チームの作業ごとの工数を整理したものを表 5-1 に示す。

[1]工数内訳は 4.3.3 節の【ステップ 2】形式記述作成・検証をさらに詳細な作業に分割してその作業ごとに要した工数である(形式手法ごとに詳細なタスクは異なることに注意いただきたい)。

表 5-1　B チームの工数内訳

チーム名	作業工数(人時)					
	Event-B 要素の抽出	リファイメント戦略の立案	形式記述の作成	形式記述の検証	形式記述の修正	合計
B1	34.0	1.5	9.5	54.5	8.0	107.5
B2	45.5	0.0	17.0	0.0	2.0	64.5
B3	54.0	0.0	16.0	20.0	0.0	90.0
合計	133.5	1.5	42.5	74.5	10.0	262.0

B1 チームから B3 チームの作業ごとの工数割合をグラフ化したものを図 5-1 に示す。

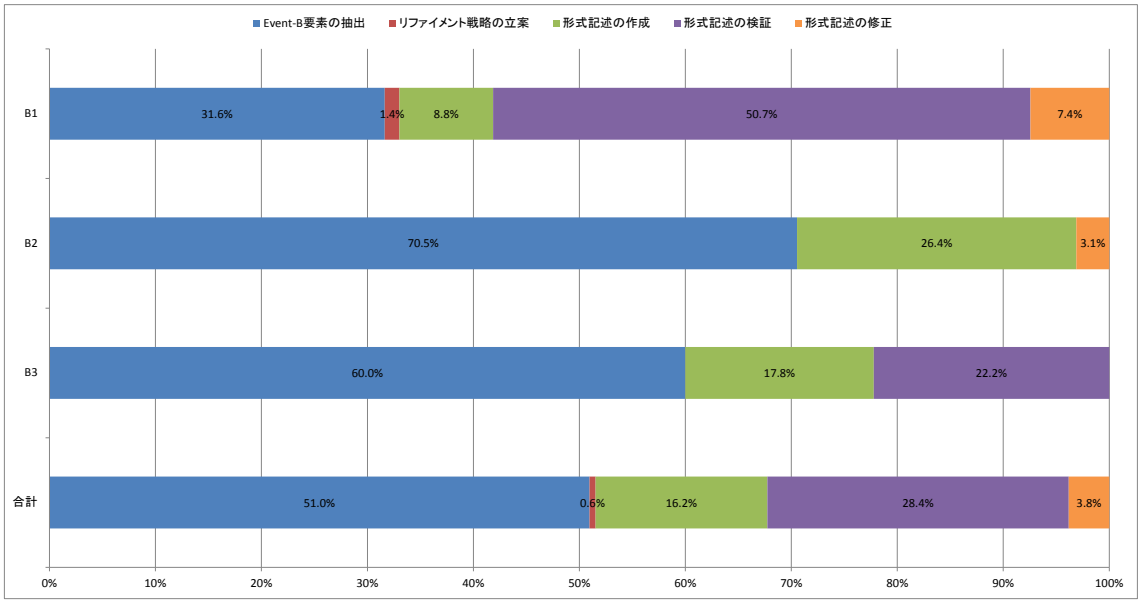

図 5-1　B1-B3 チームの作業ごとの工数割合

【S チームの工数内訳】

S チームの作業ごとの工数を整理したものを表 5-2 に示す。

表 5-2　S チームの工数内訳

チーム名	作業工数(人時)			
	情報の抽出	PROMELA 記述(LTL 式)の作成	PROMELA 記述(LTL 式)の検証	合計
S	16.5	19.5	8.5	44.5

S チームの作業ごとの工数割合をグラフ化したものを図 5-2 に示す。

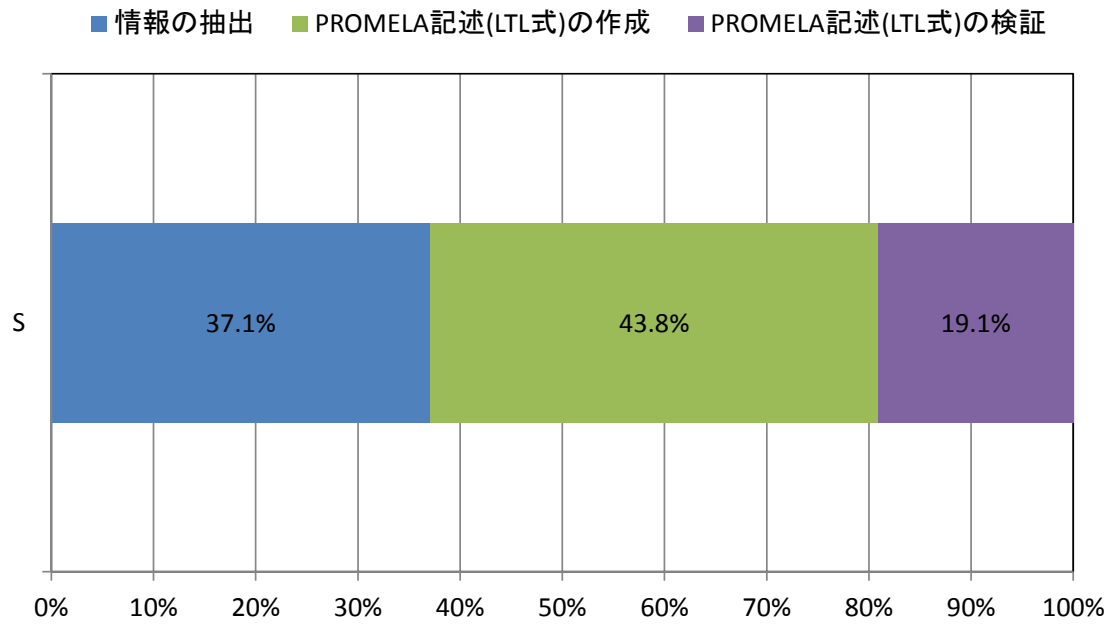

図 5-2　S チームの作業ごとの工数割合

【V チーム内の工数内訳】

V チーム各作業者に対して、作業ごとの工数を整理したものを表 5-3 に示す。

表 5-3　V チームの工数内訳

作業者	作業工数(人時)					
	既存ドキュメントの調査	形式記述作成計画立案	形式記述の作成(陰仕様)	形式記述の作成(陽仕様)	テスティング	合計
作業者 A	13.0	6.0	2.0	21.0	0.0	42.0
作業者 B	15.0	10.0	8.0	14.0	0.0	47.0
作業者 C	48.0	2.5	0.0	21.0	7.0	78.5
作業者 D	7.0	8.0	2.0	7.5	4.5	29.0
作業者 E	44.0	3.5	1.0	9.0	0.0	57.5
合計	127.0	30.0	13.0	72.5	11.5	254.0

V チーム各作業者に対して、作業ごとの工数割合をグラフ化したものを図 5-3 に示す。

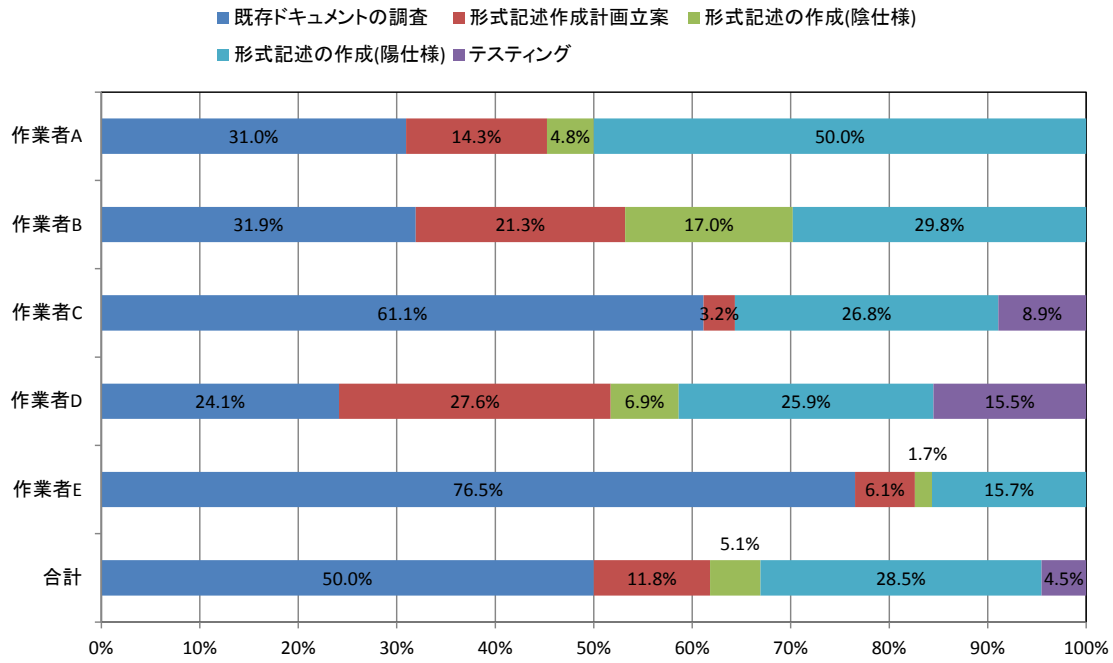

図 5-3　V チーム(作業者)の作業ごとの工数割合

5.1.2　作業効率

各実験チームのページあたりの作業効率を表 5-4 に示す。

表 5-4　作業効率

チーム名	全体工数（人時）	形式記述の作成対象ページ数(頁)	参照ページも含む総ページ数(頁)	作業効率(1)(人時/頁)※	作業効率(2)(人時/頁)※
B1	107.5	110	707	0.98	0.15
B2	64.5	106	287	0.61	0.22
B3	84.0	49	381	1.84	0.24
S	44.5	151 （重複分含む）	612 （重複分む）	0.29	0.07
V	254.0	300	700	0.85	0.36

※作業効率(1) = 全体工数 / 形式記述の作成対象ページ数

作業効率(2) = 全体工数 / 参照ページも含む総ページ数

5.1.3　指摘事項抽出効率

各チームの指摘事項抽出効率（工数単位での指摘事項数比率）を表 5-5 に示す。なお、本実験は一つの事例であること、また各チーム間で作業者のスキルや形式化した範囲が異なることから、この結果によって手法毎の優劣を考察することは妥当でないことに留意が必要である。

表 5-5　指摘事項抽出効率

チーム名	全体工数（人時）	指摘事項数（件）※2	指摘事項抽出効率（件/人時）
B1	107.5	6　(6)	0.06
B2	64.5	3　(0)	0.05
B3	90.0	1　(0)	0.01
S(1)　※1	44.5	14　(8)	0.31
S(2)　※1	44.5	20　(12)	0.45
V	254.0	31　(21)	0.12

※1　S(1)は重複分を除いた指摘事項、S(2)は重複分も数えた指摘事項で実施

※2　括弧内の数字は、指摘事項のうち「修正が必要」または「修正が望ましい」と評価された件数

5.1.4　指摘事項の分類

指摘事項の分類種別に対する各チームの指摘事項数を表 5-6 に示す。ただし、各チームで実施した対象範囲は異なるため、単純に数値を比較することは意味がない。本集計では、S チームの各作業者で指摘した指摘事項の重複分は排除して数えている。

表 5-6　指摘事項の分類

チーム名	指摘事項の分類				
	設計内容の衝突	設計内容の不足	設計内容の曖昧さ	誤字・脱字など	合計
B1	2	3	0	1	6
B2	3	0	0	0	3
B3	0	1	0	0	1
S	2	7	4	1	14
V	8	14	6	3	31
合計	15	25	10	5	55

指摘事項の分類種別に対する各チームの指摘事項数の割合を図 5-4 に示す。

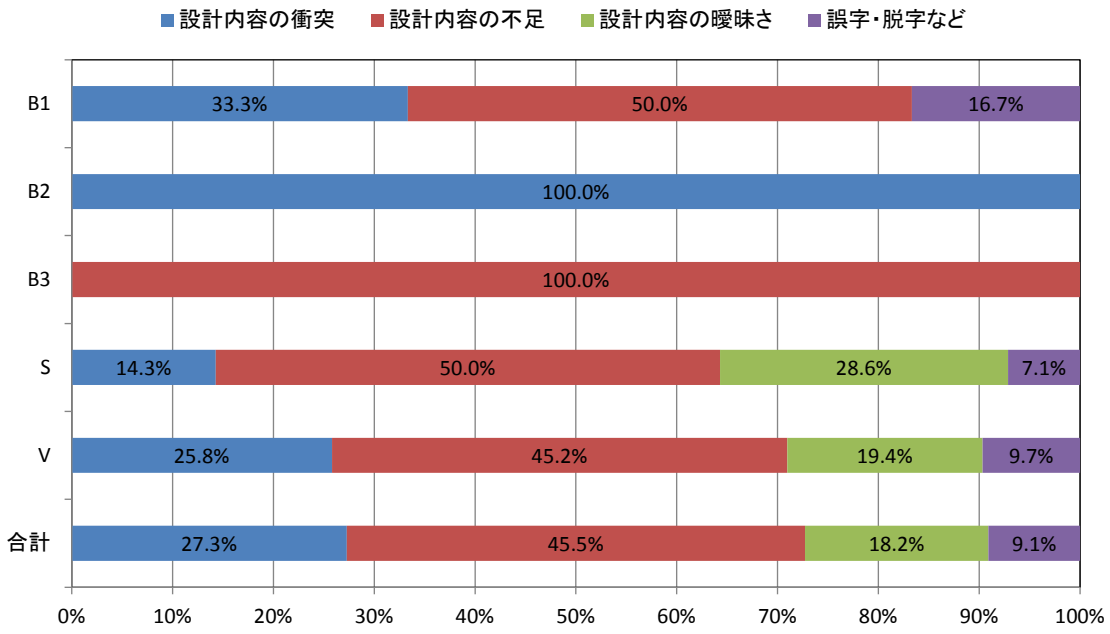

図 5-4　指摘事項の分類

5.1.5　作業と指摘事項の関係

B1-B3 チームで抽出したすべての指摘事項に対して、各作業ごとに抽出した指摘事項の分類を表 5-7、図 5-5 に示す。

表 5-7　作業と指摘事項の関係(B1-B3 チーム)

指摘事項分類	指摘事項数(件)					
	Event-B 要素の抽出	リファイメント戦略の立案※	形式記述の作成	形式記述の検証	形式記述の修正※	合計
設計内容の衝突	3	-	1	1	-	5
設計内容の不足	1	-	1	2	-	4
設計内容の曖昧さ	0	-	0	0	-	0
誤字・脱字など	1	-	0	0	-	1
合計	5	-	2	3	-	10

※「リファイメント戦略の立案」「形式記述の修正」作業では、指摘事項が抽出されることはないため、「-」とした。

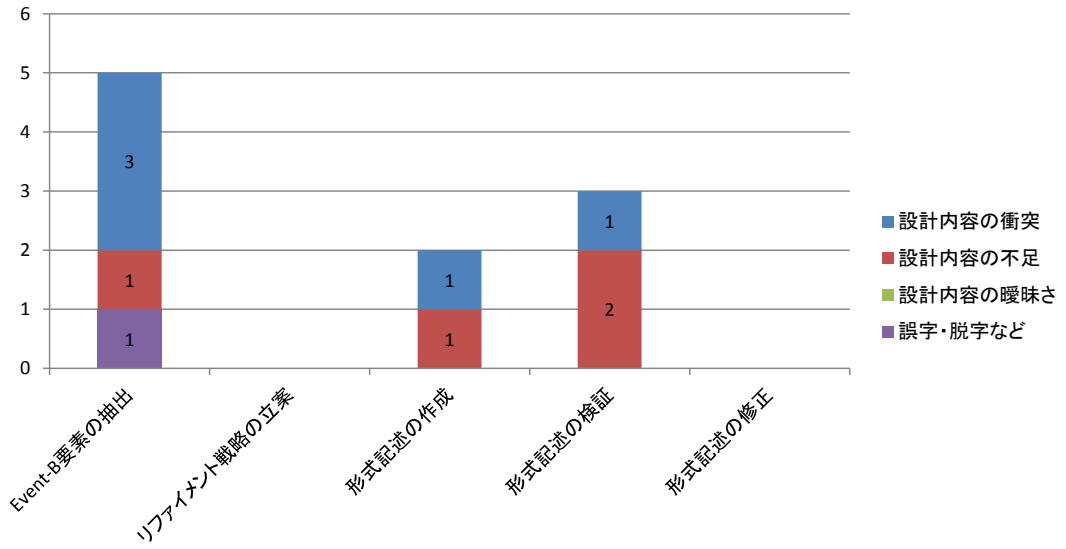

図 5-5 作業と指摘事項の関係(B1-B3 チーム)

S チームで抽出した指摘事項に対して、各作業ごとに抽出した指摘事項の分類を表 5-8、図 5-6 に示す。S チームは各作業者で抽出した指摘事項のうち、重複分も存在するが、本集計では重複分は排除して数えている。

表 5-8 作業と指摘事項の関係(S チーム)

指摘事項分類	指摘事項数(件)			
	情報の抽出	PROMELA 記述(LTL 式)の作成	PROMELA 記述(LTL 式)の検証	合計
設計内容の衝突	2	0	0	2
設計内容の不足	3	1	3	7
設計内容の曖昧さ	3	0	1	4
誤字・脱字など	0	0	1	1
合計	8	1	5	14

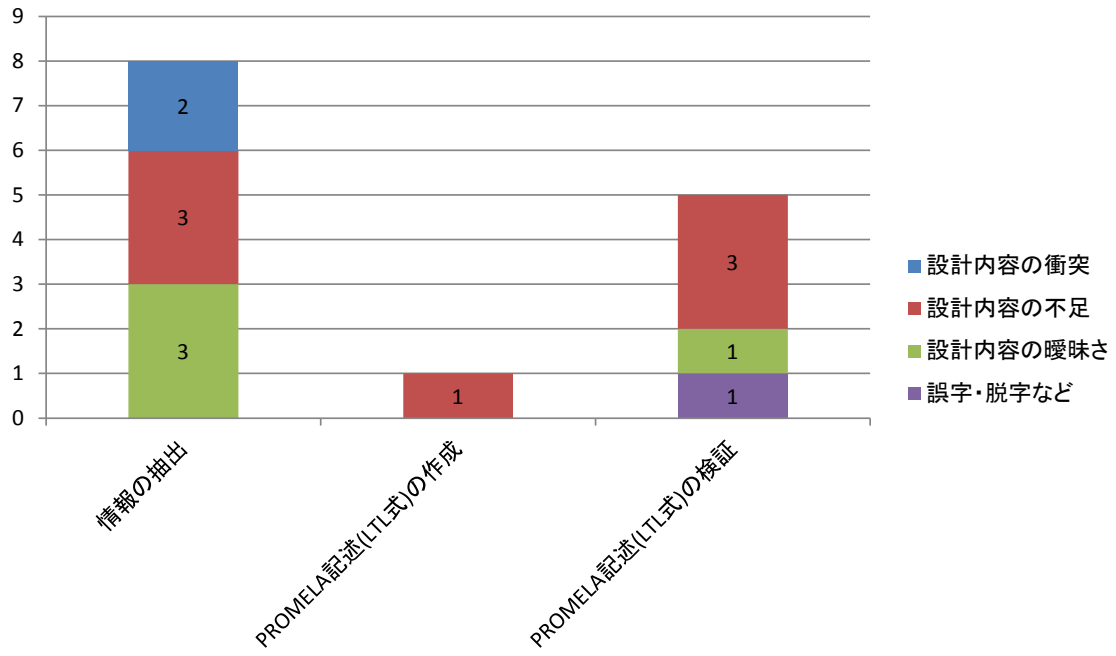

図 5-6 作業と指摘事項の関係(S チーム)

V チームで抽出した指摘事項に対して、各作業ごとに抽出した指摘事項の分類を表 5-9、図 5-7 に示す。

表 5-9　作業と指摘事項の関係(V チーム)

指摘事項分類	指摘事項数(件)					
	既存ドキュメントの調査	形式記述作成計画立案	形式記述の作成(陰仕様)	形式記述の作成(陽仕様)	テスティング	合計
設計内容の衝突	3	0	0	5	0	8
設計内容の不足	5	1	2	5	1	14
設計内容の曖昧さ	1	0	1	3	1	6
誤字・脱字など	1	0	0	2	0	3
合計	10	1	3	15	2	31

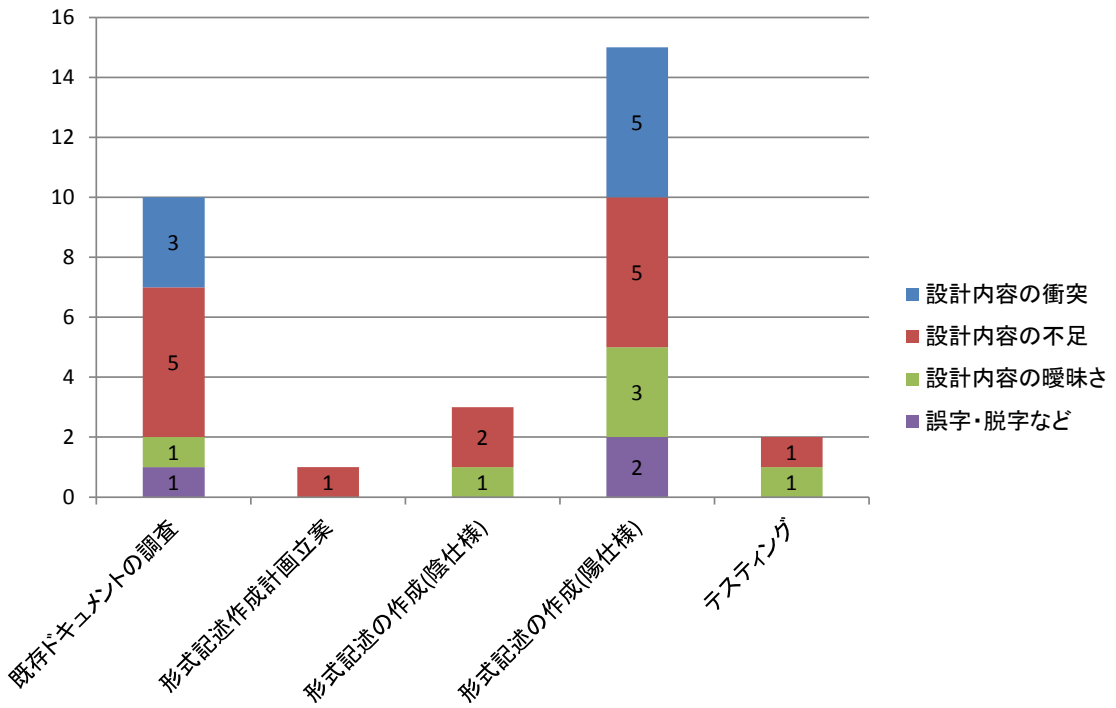

図 5-7　作業と指摘事項の関係(V チーム)

5.2 外部設計書の改善点を指摘できる効果

本実験で用いた形式手法（Event-B, SPIN, VDM++）では、形式記述の作成時に、構文・型チェックをツールで行うことにより、同音異義語や曖昧な表現といった字面上の指摘事項が抽出される。また、形式記述の検証時に、仕様アニメーション、モデル検査や定理証明をツールで行うことにより、事前・事後・不変条件違反やアサーション違反といった指摘事項が抽出される。記法および形式手法適用者が定義した形式記述と検査条件の範囲内でもれなく指摘事項を抽出できることが、形式手法の強みである。また、形式手法適用者は、ツールによるチェックで問題を起こさないという観点で外部設計書を読むため、形式記述の作成前にも、多くの指摘事項が抽出される。

本節では、本実験において抽出された指摘事項について、表 4-1、表 4-2 に示す観点からの分類結果を分析し、考察する。なお、この分類結果は絶対的なものではなく、特定の事例における傾向を示すものであることに注意が必要である。また、本節で行う分析・考察は、各実験チームが抽出した指摘事項をまとめて行うこととした。実験チームごとの分析では指摘事項の件数が少ない場合があること、形式手法の優劣比較は本実験の目的ではないことが理由である。

5.2.1 指摘事項に関するデータの整理

本実験で得られた外部設計書の指摘事項は合計 55 件であった。表 5-10 は、表 4-1 にしたがって 55 件の指摘事項を分類した結果である。

表 5-10　指摘事項の種類別件数

指摘事項の種類	件数
設計内容の衝突	15
設計内容の不足	25
設計内容の曖昧さ	10
誤字・脱字など	5
計	55

次に、表 4-2 に示した評価基準に沿って設計書提供者が判定した結果を、表 5-11、表 5-12、表 5-13、表 5-14 に示す。

表 5-11　設計書提供者による評価結果（設計書の修正有無）

評価項目	選択肢	該当件数
設計書の修正有無	修正済	13
	未修正	24
	修正対象外	18
計		55

表 5-12　設計書提供者による評価結果（設計書修正の必要性）

評価項目	選択肢	該当件数
設計書修正の必要性	修正が必要	22
	修正が望ましい	13
	修正不要	20
計		55

表 5-13　設計書提供者による評価結果（発見時期）

評価項目	選択肢	該当件数
発見時期	稼働前	20
	稼働後	0
	未知	10
計		30

発見時期については、暗黙の了解・ルールなどにより問題が起こり得ないと設計書提供者が判断した指摘については評価対象としなかった。

表 5-14　設計書提供者による評価結果（暗黙の了解・ルールに関する指摘）

評価項目	選択肢	該当件数
暗黙の了解・ルールに関する指摘	yes	29
	no	26
計		55

5.2.2　実験結果の考察

本節では、実験結果を参照し、「外部設計書の改善点を指摘できる効果」に関する考察を行う。まず、形式手法適用者が抽出した指摘事項と設計書提供者の評価による外部設計

書の修正の必要性との関係から、形式手法によって外部設計書の修正を必要とする指摘事項を抽出できることを述べる。次に、指摘事項と従来開発における発見時期との関係から、形式手法によって従来開発よりも早期に指摘事項を抽出できることを述べる。また、指摘事項と形式手法を適用する作業との関係から、形式記述の検証を行わない使い方であっても指摘事項を抽出する効果があることを述べる。最後に、指摘事項と「暗黙の了解・ルール」との関係から、形式手法適用者に「暗黙の了解・ルール」を知らしめることが、外部設計書の修正が必要な指摘事項をより多く抽出することにつながることを述べる。

表 5-15 指摘事項種別と設計書修正の必要性との関係

設計書 修正の必要性	指摘事項種別				
	設計内容 の衝突	設計内容 の不足	設計内容 の曖昧さ	誤字・脱字 など	合計
修正が必要	9 件(41%)	9 件(41%)	1 件(5%)	3 件(14%)	22 件(40%)
修正が望ましい	3 件(23%)	6 件(46%)	3 件(23%)	1 件(8%)	13 件(24%)
修正不要	3 件(15%)	10 件(50%)	6 件(30%)	1 件(5%)	20 件(36%)
合計	15 件(27%)	25 件(45%)	10 件(18%)	5 件(9%)	55 件

＊ 件数右側のカッコ内は、二重罫線で囲まれた区域の件数の総和に対する比率を表す。

形式手法適用者が抽出した 55 件の指摘事項を対象とした、指摘事項と設計書修正の必要性との関係を表 5-15 に示す。各行には設計書提供者によって外部設計書の「修正が必要」「修正が望ましい」「修正不要」と評価された指摘件数の内訳を、各列には「設計内容の衝突」といった指摘事項種別で分類した指摘件数の内訳を示す。

設計書提供者によって外部設計書の修正が必要と評価された指摘事項は 22 件(40%)ある。修正が望ましいと評価された 13 件と合わせると 35 件であり、全 55 件に対して 64%を占める。形式手法により、外部設計書の修正が必要、あるいは修正が望ましい指摘事項を抽出することができた事を示している。これらの結果から、以下のことがいえる。

［結論 1］
形式手法により、外部設計書の修正が必要な、または修正が望ましい指摘事項を抽出できた。

表 5-16　設計書修正が必要と判断した指摘事項を対象とした発見時期と暗黙の了解・ルールに関する指摘事項の関係

暗黙の了解・ルールに関する指摘事項	発見時期			
	稼働前	稼働後	未知	合計
yes	7 件(32%)	0 件(---)	0 件(---)	7 件(32%)
no	13 件(59%)	0 件(---)	2 件(9%)	15 件(68%)
合計	20 件(91%)	0 件(0%)	2 件(9%)	22 件(100%)

設計書提供者によって外部設計書の修正が必要と判断した 22 件（表 5-12 参照）の指摘事項を対象とした、発見時期と暗黙の了解・ルールに関する指摘事項の関係を表 5-16 に示す。

表 5-16 より、設計書提供者が外部設計書の修正が必要と判断した 22 件のうち、実際の開発で稼働前に発見されていた指摘事項は 20 件である。20 件のうち 7 件は暗黙の了解・ルールに関する指摘事項である。暗黙の了解・ルールは、外部設計工程時には開発に関わっていた関係者が既に認識していたことを考えると、暗黙の了解・ルールに該当しない 13 件が、外部設計工程より後に発見された改善事項であるといえる。すなわち、この 13 件は、外部設計工程時に形式手法を適用していれば抽出可能であったと考えられる。このことは、形式手法の適用により、手戻りコスト負担を削減可能なことを示唆しているといえる。

以上の分析から、本実験の結果として以下の結論が導ける。

［結論 2］
形式手法により、実際の開発では設計完了後に発見されていた改善事項を設計時に抽出できた。

表 5-17 形式手法適用時の作業と設計書修正の必要性の関係

設計書 修正の必要性	作業群			
	文書読解と 情報抽出	計画立案と 形式記述の作成	形式記述の 検証・修正	合計
修正が必要	1 件(11%)	19 件(53%)	2 件(20%)	22 件(40%)
修正が望ましい	4 件(44%)	66 件(17%)	3 件(30%)	13 件(24%)
修正不要	4 件(44%)	11 件(31%)	5 件(50%)	20 件(36%)
合計	9 件(16%)	36 件(65%)	10 件(18%)	55 件

＊ 件数右側のカッコ内は、二重罫線で囲まれた区域の件数の総和に対する比率を表す。

表 5-18 表 5-17 の作業群と各手法ごとの作業の対応関係

手法	作業群		
	文書読解と情報抽出	形式記述の作成	形式記述の検証
Event-B	・Event-B 要素の抽出	・リファイメント戦略の立案 ・形式記述の作成	・形式記述の検証 ・形式記述の修正
SPIN	・情報の抽出と中間記述の作成	・Promela 記述（LTL 式）の作成	・Promela 記述（LTL 式）の検証
VDM++	・既存ドキュメントの調査	・形式記述作成計画立案 ・形式記述の作成（陰仕様） ・形式記述の作成（陽仕様）	・テスティング

形式手法適用者が抽出した 55 件の指摘事項を対象とした、指摘事項と形式手法を適用する際の作業の関係を表 5-17 に示す。各行には設計書提供者によって外部設計書の「修正が必要」「修正が望ましい」「修正不要」と評価された指摘件数の内訳を、各列には形式手法適用者が行った作業を 3 つの作業群に分類した場合の指摘件数の内訳を示す。

実験チームが行った作業は採用した手法ごとに異なる。ここでは、作業を大きく「文書読解と情報抽出」「計画立案と形式記述の作成」「形式記述の検証・修正」の 3 つにまとめ作業群と呼ぶこととする。各手法の作業がどの作業群に対応するかを表 5-18 に示す。

全指摘事項 55 件のうち、「文書読解と情報抽出」と「計画立案と形式記述の作成」で抽出された指摘事項は 45(82%)件である。外部設計書の修正が必要と評価された指摘事項に絞ると、22 件のうち、「文書読解と情報抽出」と「計画立案と形式記述の作成」で抽出された指摘事項は 20(88%)件である。「形式記述の検証・修正」を行わないような使い方であっても、外部設計書の修正を必要とするような指摘事項を抽出できる効果があったといえる。形式化時の指摘事項抽出については、文献 [6] などの報告でも知られている。本実験においても同様の結果であったといえる。

以上の分析から、本実験の結果として以下の結論が導ける。

［結論 3］
形式記述の作成だけでも、形式手法を適用する効果があった。

開発プロジェクトには暗黙の了解・ルールがある。同じ要員が参加することにより先の開発での決めごとを文書化せずに情報共有する、あるいは、その開発で利用するプログラムフレームワーク上に機能を実装するに必要なことのみを設計書に書く、など様々な事情から暗黙の了解・ルールが生まれる。業種に固有の決まりごと(ドメインの知識)も暗黙の了解・ルールとなる。

本実験では、外部設計書の読み方といった暗黙の了解・ルールがあった。形式手法適用者には暗黙の了解・ルールは知らされなかった（設計書提供者とのQ&Aによって少数を把握できたにとどまる）。形式手法適用者が抽出した指摘事項は、所与の外部設計書のみをインプットとした場合に、形式手法によって抽出できたものである。一方、設計書提供者は、暗黙の了解・ルールを前提として指摘事項を調べ、外部設計書修正の必要性や従来開発における発見時期などを評価した。[2]

表 5-19 暗黙の了解・ルールと設計書修正の必要性の関係

設計書 修正の必要性	暗黙の了解・ルールとの関係		
	関係なし	関係あり	合計
修正が必要	15 件(58%)	7 件(24%)	22 件(40%)
修正が望ましい	8 件(31%)	5 件(17%)	13 件(24%)
修正不要	3 件(12%)	17 件(59%)	20 件(36%)
合計	26 件(47%)	29 件(53%)	55 件

＊ 件数右側のカッコ内は、二重罫線で囲まれた区域の件数の総和に対する比率を表す。

形式手法適用者が抽出した 55 件の指摘事項を対象とした、指摘事項と暗黙の了解・ルールの関係を表 5-19 に示す。各行には設計書提供者によって外部設計書の「修正が必要」「修正が望ましい」「修正不要」と評価された指摘件数の内訳を、各列には設計書提供者が暗黙の了解・ルールとの関係の有無を評価した指摘件数の内訳を示す。

指摘事項 55 件のうち、半数以上の 29 件が暗黙の了解・ルールに関係するものであった。

[2]設計書提供者側で設計書記述ルールがシステム開発時よりも本実験時点の方が厳密になったこと、設計者間では問題はないが後工程の作業者の理解容易性向上を考慮したこと等により、暗黙の了解・ルールに関係する指摘事項についても、設計書の修正が必要あるいは望ましいと評価されるものがあった。

暗黙の了解に関係しない指摘事項に限定すると、外部設計書の修正が必要と評価されたものについては 40%から 58%へ、修正が望ましいと評価されたものについては 24%から 31%へと比率が増えることがわかる。

もし、形式手法適用者に暗黙の了解・ルールを知らしめた場合、少なくとも表 5-19 に示す 26 件の指摘事項は形式手法によって抽出される。さらに、暗黙の了解に関する指摘事項の背後に潜む指摘事項も抽出されると期待される。形式手法適用者に暗黙の了解・ルールを知らしめれば、設計書の修正が必要となる指摘事項の抽出率を向上できるといえる。

開発文化の異なるオフショアに開発を発注したり、あるいは開発経験の少ない新人の参加や、プロジェクト途中での増員といった場合には、暗黙の了解・ルールはコミュニケーションの妨げとなる。第三者が設計書の検証を行うという本実験も、このような場合に当たる。暗黙の了解・ルールを発見できるのは第三者検証の効果である。

以上の分析から、本実験の結果として以下の結論が導ける。

［結論 4］
形式手法適用者に暗黙の了解・ルールを知らしめれば、設計書の修正を要する指摘事項の抽出率が向上する可能性がある。

5.3 『ソフトウェア開発データ白書 2010-2011』データとの比較

本実験は新規開発プロジェクトの外部設計書に対して行った。本比較では、外部設計は基本設計に対応するとみなしたうえで、本実験で得られた「表 5-4　作業効率」（5.1.2 節参照）、「表 5-5　指摘事項抽出効率」（5.1.3 節参照）と『ソフトウェア開発データ白書 2010-2011』[7] の「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)」（209 ページ）、「図表 8-2-3 工数あたりの基本設計レビュー指摘件数の基本統計量（1）」（207 ページ）を比較する。

本実験での作業効率(1)は 0.29～1.84(人時/頁)、『ソフトウェア開発データ白書 2010-2011』の P75 値 1.166(人時/頁)である。これより、１チームを除いた本実験の作業効率は、『ソフトウェア開発データ白書 2010-2011』の 75%のプロジェクトに収まることが分かり、通常レビューの作業効率と比較しても大きな差はないことが分かる。

表 5-20　作業効率(表 5-4 抜粋)

チーム名	作業効率(1) (人時/頁)※	作業効率(2) (人時/頁)※
B1	0.98	0.15
B2	0.61	0.22
B3	1.84	0.24
S	0.29	0.07
V	0.85	0.36

※作業効率(1) = 全体工数 / 形式記述の作成対象ページ数

作業効率(2) = 全体工数 / 参照ページも含む総ページ数

【参考】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）[人時/頁]

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

【出典】ｿﾌﾄｳｴｱ開発データ白書 2010-2011 [7]

(注意) 図表 8-3-1 は標準偏差が大きく、プロジェクトごとのばらつきがかなり大きいため、単純な平均値との比較は実施しない。

また、本実験での指摘事項抽出効率は 0.01～0.45(件/人時)、この値は『ソフトウェア開発データ白書 2010-2011』の基本設計での平均レビュー指摘件数 1.647(件/人時)の 0.6～27.3%であり、通常レビューと比較すると指摘事項抽出効率はかなり低い。しかし、これは本実験がすでに通常レビュー実施済みの外部設計書を対象としたことを加味しなければいけない。逆に通常レビューで抜け漏れた指摘事項が形式手法を適用することにより、上記の指摘事項抽出効率で指摘事項を抽出することができるとも言える。

表 5-21 指摘事項抽出効率(表 5-5 抜粋)

チーム名	指摘事項抽出効率 (件/人時)
B1	0.06
B2	0.05
B3	0.01
S(1)※	0.31
S(2)※	0.45
V	0.12

【参考】図表 8.2.3 工数あたりの基本設計レビュー指摘件数の基本統計量[件/1,000 人時]

N	最小	P25	中央	P75	最大	平均	標準偏差
52	0.000	231.337	1,126.603	2,333.333	11,156.250	1,646.550	2,011.253

【出典】ソフトウェア開発データ白書 2010-2011 [7]

6. その他に実施した実験

当 WG では、上述した実験のほかに、形式手法の効果をより明らかにするための参考とする目的で、その他の実験も行った。本章では、その他に実施した 2 つの実験について述べる。

6.1 システム稼働後の問題を事前発見できる効果の確認

表 5-13 に示す結果のとおり、本実験では、実験対象としたシステムの稼働後に発見された改善事項の指摘はなかった。しかしながら、指摘事項のなかに「稼働後に発見された改善事項」が含まれなかったことは「たまたまの結果」であり、形式手法の効果を限定するものではないというのが当 WG で議論した見解である。4.3.2 節で示したように、各実験チームは設計書の一部を切り出して形式手法を適用する対象とした。形式化の対象とした範囲に、「稼働後に発見された改善事項」がたまたま含まれなかったことが今回の実験結果の原因として考えられた。

そこで、当 WG の見解を裏付けるために、形式手法適用者に予め「稼働後に発見された改善事項」が含まれる範囲として設計書のページ番号のみを伝え、形式手法適用者が既に実施した実験と同様の方法で形式化と検証を行う実験を実施した。

実験内容の詳細は別冊「詳細報告書」第 6 章に報告するが、結果として、他の指摘事項を抽出したのと同様の方法で「稼働後に発見された改善事項」を発見することができた。

この結果により、今回の実験で採用したような「外部設計書の検査に形式手法を適用する方法」によって、システムの稼働後に修正が必要となるような改善事項を事前に発見し得ることを確認できた。

6.2 形式手法の表現力と検証能力の確認

高い信頼性が要求されるシステムでは、設計に誤りがないことを設計書のレベルで検証できていることが必須になりつつある。機能安全規格 IEC61508 やセキュリティ規格 IEC15408 などの国際規格では、このような検証を可能にする手段として形式手法を推奨し

ている。しかし、国内での形式手法の適用例はまだ限られており、実システムへの適用に必要な基礎データの蓄積が十分であるとは言い難い。

そこで、設計書レベルの検証における形式手法のフィージビリティを調査するための実験を実施した。具体的には、外部設計書に記載されたアクセス権階層とアクセス権設定／剥奪処理の両者を形式仕様言語を使って記述するとともに、後者によって前者に違反するアクセス権が設定されないことを、形式的に検証した。実験の目的は、この作業を通じて以下を確認することである。

- 対象とする設計書の必要な部分が、形式仕様言語によって表現できること
- 記述した形式仕様の整合性を確認するために必要な検証が、十分な厳密性のもとに行えること
- 上記の記述と検証が、一般的な計算機環境の下で妥当な作業時間内に行えること

ここで、与えられた外部設計書の正しさの直接的な保証は、実験の目的ではないことに留意が必要である。与えられた外部設計書は保証を意図して作成されたものではなく、保証のための評価に適した厳密性を持つものではない。この実験の目的は、保証が必要となる場合のための基礎データを収集することである。

実施した実験内容の詳細は別冊「詳細報告書」第 7 章に報告するが、結果として以下のことを確認できた。

- 外部設計書で 28 頁にわたって記述された複雑な仕様を、257 行の Event-B で明確に記述できた
- この仕様を記述する上で、Event-B の記述力が不足することはなかった
- ツールによって生成された 182 項目の証明課題を全て証明できた
- 設計書の理解、Event-B の記述、および検証の全ての作業を 45 人時で実行できた

この結果により、以下のことが結論付けられる。

- 対象とする設計書の必要な部分が、形式仕様言語によって表現できる
- 記述した形式仕様の整合性を確認するために必要な検証が、十分な厳密性のもとに行える
- 上記の記述と検証が、一般的な計算機環境の下で妥当な作業時間内に行える

7. 実験結果の評価

実験で得られた結果に対して、WG に参加したユーザ企業の委員および学識経験者の委員がそれぞれの視点で評価した。

7.1 ユーザ企業の委員による評価（1）

住友電気工業(株)では、システム開発のＱＣＤ改善に継続的に取り組んでおり、外部設計の品質改善が１つの改善テーマになっている。1995 年から佐藤正美氏が考案したＴ字形ＥＲ手法を全社に導入し、システムの構造を図面化することで品質改善の成果を上げている。

また、最近では要件と機能のトレーサビリティを確認するツールを開発し、システム・オーナーに対する品質も改善を進めている。今後の課題はシステムの振る舞い（ロジック）に関する品質の保証であり、その解決策として形式手法に注目している。た

形式手法のエンタープライズ系システムへの適用に際しては、わからないことが多く、今回の実証実験では以下の点に注目してきた。

(1) 形式手法には、VDM++, Event-B, SPIN 等いろいろな手法があるが、ビジネスシステムでは、どの手法がどのような目的、対象で有効か
(2) 形式手法導入の為の追加工数はどの程度か
(3) 形式手法の検出プロセスの性能(検出効率、見逃し率等)はどの程度か

これらの視点で今回の実験結果について考察する。

１．手法の選択

それぞれの手法にはそれぞれの特徴がある。例えば SPIN は複数のプロセスが並行して実行された際に不具合が発生しないかの検証に適している。この特徴は、組み込み系システムでは多くの活用シーンが考えられるが、ビジネスシステムでの適用シーンは限定的であるかもしれない。現在のところ、各手法を体得していないユーザー企業の開発者が最適な手法を選択する事は難しく、コンサルタントの支援が必要であるというのが本ＷＧの共通認識である。

２．導入時の追加コスト

表 5-4 によると形式手法による検証コストは、0.29～1.84MH/ページとなっている。弊社では、外部仕様書が HTML で表現されている為、ページ数の評価は実感に合わないが、外部設計と並行して進めるのであれば、導入可能な工数だと考えられる。ただし、手法により検証の対象が異なり、各ページに記載された全ての内容が検証される訳ではないので、補助的なものと位置づける必要がある。

３．検出プロセスとしての性能

今回の実験では、合計 560.5MH の工数で 55 件の指摘が出ている。理解しやすいように 1MM = 150MH とすると、１人月あたり 14.7 件の指摘があり、数値的には効果があると言

える。以下、形式手法を導入しようとしているユーザー企業の視点でもう少し内容の分析を行う。

形式手法では、(a)形式記述を作成する為に、仕様を理解する工程と(b)作成した形式記述を入力とするツールによる検証の工程に分けることができる。形式手法に対する期待は(b)で、人間の配慮が行き届かないケースを自動的に検証し、不具合を指摘してもらうことであった。今回の実験では、55 件のうち２件(表 5-17)が(b)に相当する。期待に対して少ない件数であった。また、この２件は実際にはトラブルが発生しない指摘であった。

次に検出効率の考察を行う。表 5-19 を参照すると、修正が必要な指摘（「暗黙のルール」と関係なしに限定）は 15 件であり、１人月あたり 4.0 件見つかっている。手法別で集計すれば、0.57～6.68 件/人月の範囲でばらついているが、レビュー済みの外部設計書の検証プロセスとしては効果の期待できる値が出ている。ただし、その多くは形式記述の作成の為の仕様の理解の中で見つかっており、形式手法の特徴による効果かどうかは判断が難しい。

例えは、弊社ではＴ字形ＥＲ手法の作成により各工程の品質が向上し、開発コストが外部設計からＩＴまでの全工程が約３割削減できている。今回指摘された暗黙の了解、区分、コード値の不一致等の指摘は、用語や区分値の定義、業務のサブセット分析をきっちり行えば、かなり防止できると思われる。

まとめ

今回の実証実験では、定量的なデータが収集できた点で成果があった。しかし、一方で検出された欠陥の内容を見ると期待した程の効果は得られなかった。要因としは、

- 今回の対象システムが形式手法に向いていなかった
- 限られた期間・工数の中で実験であったため、形式手法の能力が十分発揮できなかった

等が考えられる。

形式手法はスキル獲得のハードルが高いという欠点を持っており、導入を考えている企業にとっては導入コストに見合う予想効果を納得した上で実施したい。形式手法の普及の為には以下の課題に取り組んでいく必要がある。

(1) ビジネスシステムにおける各種法の成功例の蓄積
(2) 検証すべき性質の設定ノウハウの蓄積
(3) 形式手法の導入に適した外部仕様書の構成の検討
(4) 欠陥検出能力の改善（欠陥の見逃し率の削減等）

（住友電気工業株式会社　中村伸裕委員）

7.2 ユーザ企業の委員による評価（2）

東京証券取引所では、“上流工程の品質がシステム全体の品質を決定する”との認識のもと、特にユーザ側の責任である要件定義書（外部設計を含む）の品質向上に取り組んでいる。今回の実証実験においても、上記取組みの１つのアプローチとして形式手法が有効か否かという観点から評価を行った。

一定の効果が認められると評価したのは、以下の２点である。

（１） 実際のプログラム上の不具合を２件、摘出できた点

１件目はこれまで検出できていなかった不具合、２件目はテスト工程において検出したが、実際の業務運用においては問題がなかったため、修正せずに仕様追認したものである。

上記２件の指摘事項については、従来のレビューにより検出できないというものではないが、時間をかけて読み込みを行わないと検出できないレベルの不具合であり、この不具合を形式手法により、比較的短時間で検出できたことは注目してよい。このことは、追加実験の結果からも明らかであり、実際には稼働後に発見した不具合をわずか数時間で発見している。

（２） 設計上考慮されていなかった観点を指摘していただけた点

具体的には、「データベースのメンテナンス時のミスや仕様変更時の影響範囲の確認ミスにより、複数のデータの組合せが不正な状態になり、予期せぬトラブルを引き起こす可能性がある。これを予防するためのガード条件を設けるべき」という指摘（別冊「詳細報告書」3.5 節参照）となる。

この指摘については、本システムが求める信頼性の要求レベルが、こうした事態にまで予め備えておくべきであるというほど高くないために、スコープ外となってしまったが、仮に形式手法を採用していれば、こうした観点についても設計工程で考慮できた可能性があったということは評価できる。

一方、指摘事項の大半が、発注者サイドからするとあまり有効な指摘ではなかったのも事実である。特に設計書の記述ルールを実験者が知らないことに起因する指摘が多かった（今回の実験では「暗黙の了解・ルール」に分類している）。

また、発注者サイドとしては、従来のレビューによる不具合の摘出と比較してどちらがどれだけ効果的であるかが気になるところであるが、実際にレビューで摘出した不具合が形式手法でどれだけ摘出できたか、それがレビューと比較してどれだけ効率的であったかの検証はスコープ外であったために明らかにされていない。

このあたりが明らかになり、最終的には、従来のレビューによる方法に比べ QCD がどれだけ改善するかを検証することが今後の課題といえよう。

結論としては、一定の効果は認められるものの、その網羅性、効率性については実験のスコープ外であり検証できていないため、総合的な評価はできない、と言わざるを得ない。ただし、外部設計書等の品質を向上させるための方策のひとつとして、従来のレビューに加えて形式手法を採用することは十分可能であると思われる。

（株式会社東京証券取引所　古川正伸委員）

7.3 学識経験者による評価（1）

まず、今回の実験は、数ある形式手法の中から三つの異なる手法を適用してそれぞれの結果を示すことによって、形式手法に対する感触と形式手法全体の中での方向感を与えたものとして実用的な観点から貴重なものであると評価できる。今回の実験において具体的に形式手法適用のプロセスを体験したことにより示された工数割合に関する複数のデータによって、厳密なシステムの記述のために必要なドメインに関する理解とそれに基づくモデル化に要した時間と労力に見合う成果が得られることを明らかにした点は評価できる。

今回の実験では、評価項目および基準を提示して、それに基づいて考察を行った。実験の対象となった題材や実験の実施方法に関しては、完成後の外部設計書を対象としてどれだけの不具合を検出できるかということを主な目的としたために、形式手法の有用性についての限定的な評価にならざるを得なかったにせよ、一つの具体的な結果を示したこと自体に価値がある。今回の実験をきっかけとして、今後は、より多種多様な対象ならびに適用方法に対して様々な観点からの経験知見が蓄積され、共有されることが望まれる。

形式手法を用いたことにより何がどれだけ改善されるかに関する定量的な評価がしばしば要求されるけれども、システム開発は形式手法だけで完結して遂行できる訳ではない。それでも、システム開発プロセス全体において形式手法に基づくシステムの記述と分析／確認がどのような効果をもたらすかに関する実感が今回の実験で得られたものと思われる。5.2.2 節に掲げられた 4 つの結論は、これまでにも様々な事例報告において既に述べられてきたものである。しかしながら、そのこと自体が、形式手法の適用を実際に経験すれば、形式手法のこれらの一般的な有用性を実感できるということを意味している。これが、本報告書を通じて我が国のシステム開発の現場に伝わって、自分達も実際に形式手法を導入してみようという機運が高まることを期待する。

今回の実験では、外部設計書の検査に形式手法を適用した場合にどのような効果があるのかを明らかにすることが主要な目的の一つとして設定された。言うまでもなく、形式手法適用の目的ならびに適用形態は、個々のシステム開発毎にそれぞれ異なる。また、前述のように形式手法だけで完結してシステム開発を遂行できる訳ではない。開発の現場にお

ける従来のシステム開発プロセス全体の中にうまく形式手法を組入れて、既存の手法と連携して形式手法（の精神）に基づく効果を活用することが肝要である。今後は、我が国においても、もはや実証実験ではなく、実システムの開発において形式手法適用の実践を地道におこなって、その経験および知見を共有し再利用できる体制が整えられることを期待する。

（九州大学　荒木啓二郎委員）

7.4 学識経験者による評価（2）

【適用実験の概要と評価】

今回の実証実験は、形式手法として、Event-B、VDM++、SPIN を選び、主として、「検査」の使い方（3.1 節参照）を試みたものである。

選択した形式手法は、いずれも、技術情報ならびに無償ツール利用可能という 2 つの面で、取組みやすいものである。一方、各々の技術発展から考えると、Event-B と VDM++は、そもそも「構築」の使い方（3.1 節参照）を想定した考案されたといえる。また、今までに公開されている適用法から考えると、Event-B は要求モデリング等の上流工程からの使い方が多い。VDM++は処理アルゴリズムが明確になった工程で良い成果が出ているようである。共に汎用の手法であることから、「何でなければならない」というものではないが、公開事例からの観測によって述べた。一方、SPIN は「検査」としての利用法が多い手法である。SPIN は振る舞い仕様と呼ぶ観点からの記述と解析に特化することで自動検証可能な使いやすい手法にしているというが特徴がある。「構築」としての使い方も可能であるが、適用範囲が振る舞い仕様に限定される。

各手法を「検査」の目的で用いる際の考慮点を整理する。与えられた外部設計書に SPIN を適用する際には、振る舞い仕様が重要な役割を果たす箇所を選べば良い。一方、汎用な形式手法（Event-B や VDM++）は、柔軟性が高い（工夫によって様々な観点からの表現と解析ができる）ことから、適用箇所を考察する段階ならびに細かな技術面での綿密な調査が大切である。このことは、同時に、適用の難しさを示すことでもある。

さて、実証実験から得られた実際の効果報告によると、

- 形式化の作業で不具合の発見数が多い
- 解析の作業では、計算の仕組みが絡むような不具合がみつかる

ということであった。これは、既に世の中で知られている知見（たとえば、文献 [6]）と一

致する。大きな驚きがない反面、実証実験が適切にかつ着実に実施された証拠ともいえる。

【今後への期待】
3.1 節にあるように、形式手法の使い方は大きく分けて 2 種類がある。どちらの方法が正しいというものではない。しかし、どちらの使い方をするかによって、適用の勘所、適用ノウハウが異なる。今回できなかった「構築」作業に関わる事例を得て、ガイドラインを整備していくことが期待される。

また、本年行った「検査」の作業を効果的に行うことを目的とした「理想的な仕様書構成」のガイドラインも重要である。従来型の仕様書は、形式手法を援用した検査を意識していないことから、仕様書を読み解く作業に工数がかかっている。形式手法の適用を前提とした仕様書の構成法があきらかになれば、本年度の作業が容易になる。さらに、このような仕様書は、「構築」作業の結果を、従来型仕様書に位置付ける際にも効果があると期待される。

本年度とは異なる観点からの実証実験を行って、さらに、適用経験を積み、その成果を集積することが重要である。今後、形式手法適用に関わる知識の体系を整理し、産業界に提示していくことが期待される。

（国立情報学研究所　中島震委員）

7.5 学識経験者による評価（3）

・外部設計書のレビューで形式手法を利用することについて、従来方法と比較して優れた点は認められたか

形式手法をエンタープライズ系システムの外部設計書のレビューに適用できることを実証した点で画期的な成果である。また実際に、外部設計書の修正点を摘出できた点は評価できる。

形式手法によって摘出された修正点の原因は、日本語による外部設計書の曖昧さにある。したがって、形式手法でなければ今回の修正点が摘出できなかったのかということについては、結論を保留すべきである。また、形式手法を適用した結果、修正点がすべて摘出できたかどうかについても議論する必要がある。

なお、従来方法と、同じ条件で、形式手法を適用して比較した結果ではないことに留意

する必要がある。

・実際に、外部設計書のレビューに形式手法を利用することを実践するための課題は何か

外部設計書のレビューに形式手法を適用する上で、次のような課題がある。

A)必要とされる形式手法の専門知識の水準と成果（摘出事項）の関係

どの程度の専門知識があれば、どのような成果が期待できるのか？形式手法の習熟度と成果には関係があるのか、ないのか？

B)形式手法の適用箇所の選択

外部設計書全体に適用するのか。部分適用だとすると、どのような部分に適用するのがいいのか？

C)形式手法を適用するための外部設計書の完成度

完成度が低い方がいいのか、完成度が高い方がいいのか。完成度が高ければ形式手法を適用しても修正点を摘出できない。完成度が低すぎれば、形式手法の適用が困難になる。

D)形式手法によるレビュープロセスの整備

どのような手順で形式手法を外部設計書のレビューするのか？従来方法によるレビューと独立に実施するのか、順番にレビューするのか、従来方法によるレビューを実施しないのかなどについて、効果を明確にするとともに、推奨手順を整備する必要がある。

（名古屋大学　山本修一郎委員）

8. まとめ

当 WG で実施した実証実験では、外部設計書を形式仕様言語による記述に変換し、ツールを利用して検査する、という形式手法の適用法を実施した。実験の結果、形式手法適用者の観点による指摘事項が 55 件抽出された。これらの指摘事項には誤字・脱字のような表記上の改善点や、論理的な記述の改善点など、様々なものが含まれたが、設計書提供者の観点で「設計書の修正が必要である」と評価された指摘事項が 22 件あり、実際の開発に役立つ可能性を確認できた。また、形式手法を適用する作業毎の工数も実測でき、当 WG で想定した実証実験の目的を達することができた。

今回の実験は、形式手法の適用者が設計書の内容を全く知らない状態から、4 か月余りで形式化し検証するという期間設定のもとで実施したため、実験で確認できることが限定された。例えば、実際の開発で問題となった不具合の内容をあらかじめゴールに設定し、「この不具合を形式手法で発見できるか」といった確認はできなかった。言い換えれば、「形式手法によってこれだけの改善事項を発見できた」という結果は得られたが、逆に、形式

手法で発見できないことの明確化はできなかった。また、7.2 節でコメントされているように、形式手法を用いない場合との定量的な比較まではできなかった。

今後は、今回の実験で確認できなかったことも含めた実証実験が望まれる。例えば、7.4 節で示されているとおり、今回の実験は外部設計書の「検査」に形式手法を利用する方法を試みたものであったが、「構築」に形式手法を利用した場合の具体的な手順や定量的効果を明らかにすることが望まれる。また、今回の実験で外部設計書を形式化する過程で「形式化するために、外部設計書に書かれているべきこと」に関する知見が得られたケースもあった（別冊「詳細報告書」5.5.2 節参照）。形式手法の適用法の検討が、外部設計書や要件定義書などをいかに書くと良いか、という指針につながる可能性がある。

今回の実験は、形式手法の効果や工数を明らかにする目的で実施したものであったが、ある観点での効果や工数が実証され、一定の成果をあげられたと考える。今後さらなる実証実験が行われ、形式手法の効果や適用コスト等がより明らかとなり、開発現場で適切な活用が進むことのきっかけとなれば幸いである。

参考文献

1. 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター. 機能要件の合意形成ガイド. (オンライン) 2010 年. http://sec.ipa.go.jp/reports/20100331.html.

2. 中島　震. ソフトウェア工学の道具としての形式手法. : NII Technial Report, 2007.

3. Dependable Software Forum. 形式手法適用手順（Event-B 編）【改訂版】. Copyright © 2011 Dependable Software Forum, 2011.

4. Dependable Software Forum . 形式手法適用手順（SPIN 編）【改訂版】. Copyright © 2011 Dependable Software Forum, 2011.

5. Dependable Software Forum. 形式手法適用手順（VDM++編）【改訂版】. Copyright © 2011 Dependable Software Forum, 2011.

6. Steve Easterbrook, ほか, 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.

7. 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター. ソフトウェア開発データ白書 2010-2011. 2010.

8. ジョン・フィッツジェラルド、ピーター・ゴルム・ラーセン、ポール・マッカージー、ニコ・プラット、マーセル・バーホフ　（翻訳：酒匂寛）. VDM++によるオブジェクト指向システムの高品質設計と検証. : 翔泳社, 2010.

執筆者一覧

（主査、副主査、事務局を除き、五十音順）

主査	塚本　英昭	株式会社エヌ・ティ・ティ・データ
副主査	橋本　祐介	日本電気株式会社
	荒木　啓二郎	国立大学法人九州大学
	櫟　　粛之	日本電信電話株式会社
	植木　雅幸	SCSK 株式会社
	上原　忠弘	株式会社富士通研究所
	梅村　晃広	株式会社エヌ・ティ・ティ・データ
	大坪　稔房	株式会社日立製作所
	小川　千之	SCSK 株式会社
	奥山　史隆	株式会社東京証券取引所
	河本　孝久	富士通株式会社
	菊地　英幸	株式会社富士通研究所
	來間　啓伸	株式会社日立製作所
	佐原　伸	SCSK 株式会社
	鷲見　毅	株式会社東芝
	高田　沙都子	株式会社東芝
	只野　完二	株式会社日立製作所
	田端　一也	株式会社エヌ・ティ・ティ・データ
	中島　震	大学共同利用機関法人情報・システム研究機構 国立情報学研究所
	中村　伸裕	住友電気工業株式会社
	長谷川　哲夫	株式会社東芝
	平田　貞代	富士通株式会社
	古川　正伸	株式会社東京証券取引所
	宗像　一樹	株式会社富士通研究所
	山崎　雄大	日本電気株式会社
	山本　修一郎	国立大学法人名古屋大学
事務局	向山　輝	情報処理推進機構技術本部ｿﾌﾄｳｪｱ･ｴﾝｼﾞﾆｱﾘﾝｸﾞ･ｾﾝﾀｰ