

CFD 法の導入による基本仕様の正確性の向上

- 取り入れるテスト技法の選択と評価 -

Improvement of accuracy by introducing the CFD method for basic design

- A study to adopt and evaluate testing method -

主査 : 奥村 有紀子 (有限会社デバッグ工学研究所)
副主査 : 堀田 文明 (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)
研究員 : 忍田 修 (A J S 株式会社)
鈴木 康弘 (株式会社東京ビジネスソリューション)
高辻 洋平 (株式会社インテック)

研究概要

基本仕様書において仕様を文章で表現した場合、仕様が読み手に誤解を与える恐れがある。また、図や表を使って仕様を書くときに、機能の実行条件を考慮しないと、機能要件を満たすために必須の仕様要素の記載が漏れる可能性がある。このような基本仕様の不備は、詳細設計やテスト設計などの後工程の品質低下と、手戻りや作業追加による開発コスト増大の一因となる。本研究で我々は、基本仕様における機能仕様の不正確を防ぐために、基本設計にテスト技法を取り入れることにした。SQiP 研究会の分科会活動を通じてテスト技法について学習した結果、CFD 法を取り入れることで組み合わせの仕様を正確に記載できると考えた。また、実際のプロジェクトにおいて効果を実証するとともに、基本設計に適用する上で実用的な工夫を試みた。

Abstract: Textual expressions of specification in basic design document may lead successor's misunderstanding against designer's original idea. Also, figures and charts without consideration of functional conditions sometimes lack mandatory factors to meet required specifications. Such deficiencies in basic design document can be one of the factors to degrade quality of post processes, enforcing unscheduled tasks to waste development resources. In this study, we decided to introduce software test methods into basic design document. As the result of studying various methods, we considered CFD method as the best way to improve defects mentioned above. We also evaluated the effect and made practical arrangement to apply for basic design in this study.

1. はじめに

本研究は、ソフトウェア開発のプロセス^[1]の上流で作成する、基本仕様書の正確性を向上する手法の提案である。基本仕様書の中で条件の組合せを文章で表現すると、表現が曖昧であったり、条件網羅に対する考慮が不十分であったりすることにより、仕様が読み手に誤解を与える恐れがある。仕様を正確に表現するため、デシジョンテーブルなどの図表を使って組合せ条件を記述することは従来より行われてきた^[2]。しかし、単に図表形式で表現するだけでは、条件および結果を整理する過程で、有効系（仕様として定める処理の流れ）や無効系（エラー処理など、仕様外とする処理の流れ）の洗い出しが不十分となる恐れがある。これらの基本仕様の不備が原因で、詳細設計やテスト設計などの後工程における成果物の品質が低下し、予期せぬ作業の手戻りや追加作業を余儀なくされる。

本研究の目的は、基本設計を行う際に、テスト技法を設計に取り入れることで、基本仕様の正確性を向上することである。その手段として、テスト技法の一つである CFD 法を用い

第5分科会（テスト技法グループ）

て的確な組合せ条件を生成し、デシジョンテーブルで表現する手法を、基本設計に取り入れることを提案する。

手法の効果は次の手順で確認した。

- (1) 研究員の所属企業で過去に作成された基本仕様書にある、文章で表現された機能仕様について、CFD法を用いてデシジョンテーブルを作成し、網羅性を比較した。
- (2) 既存／新規両方の基本仕様書を元に総合テスト設計書を作成して、作成されたテストケースの内容と作業時間を比較した。

この結果、以下のことが分かった。

- (1) 組合せ条件の網羅性は、既存のテスト設計よりも高くなった。
- (2) テスト設計にかかる工数が削減できた。

本論文の構成は次のとおりである：

「2. 背景」では、本研究の動機となった、研究員の所属企業におけるソフトウェア開発上の課題について説明する。「3. 提案」では、基本仕様書に取り入れるテスト技法としてCFD法の採用を提案する。「4. 評価と考察」では検証を行い、結果とその詳細を示す。「5. 結論と考察」ではこの論文の結論と、この論文で未解決の課題について述べる。

2. 背景

2.1 開発プロジェクトで発生するトラブルの根本原因

本研究に先立って、下流工程の品質に影響する上流工程のフェーズを調査した。図1は研究員が所属する企業で実施された新規開発プロジェクト1件において、総合テストで発生したトラブル（総数105件）の根本原因を分類・集計した結果である。図1から、このプロジェクトで発生したトラブルのおよそ7割は設計不良に起因することが分かった。

また、図2は図1の設計不良の原因を詳細に分類した結果である。この結果から、設計不良のうちおよそ4割は機能条件設計がトラブルの要因となっていることが分かった。

機能条件設計はトラブル全体のおよそ3割程度であるが、これを改善することで、設計不良の最も頻出するトラブルを防止できると考え、本論文のテーマとした。

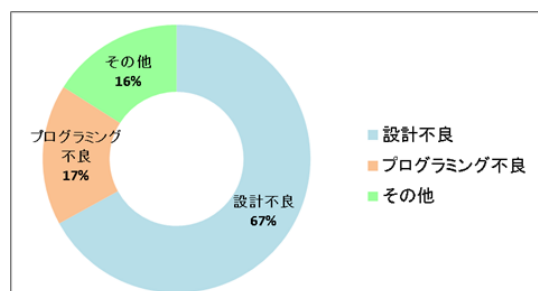


図1 総合テストで発生したトラブルの原因分析

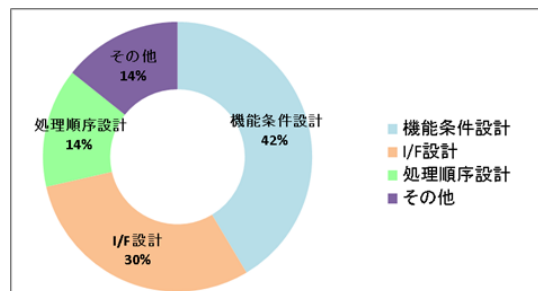


図2 設計不良の詳細な分類

2.2 基本仕様に見られる不備

ここで、実際の基本仕様書で見られた文章や図表による表現を取り上げ、曖昧な仕様や仕様の漏れの例を指摘する。図3は、研究員の所属企業で作成された基本仕様書の、文章表現による仕様の例である。この例に挙げた基本仕様には次の問題がある。

- (1) 条件と処理が「(3)機能要件」の1箇所にとまらされておらず、それ以外の項目を熟読しないと機能を理解できない。
 - 例えば入出力条件に「国内拠点は JPY、ローカルを日本円とする」とあるが、機能要件にも拘らず機能要件以外の箇所を読まないと把握できない。

第5分科会（テスト技法グループ）

<p>(1)概要説明</p> <ul style="list-style-type: none"> 海外拠点についてローカル通貨にて保持されている取引明細データを現地通貨に対応したレートによって日本円へ換算処理をおこなう。 <p>(2)前提条件</p> <ul style="list-style-type: none"> 夜間処理としてジョブにて実行される。 正常終了はログを出力し後続処理を実行、異常終了の場合はエラーログを出力し後続処理をおこなう。 夜間 12 時に時間起動で実行する。 起動時はメンテナンスモードとし、通常運用を停止する。 <p>(3)機能要件</p> <ul style="list-style-type: none"> 国内拠点は換算しない、海外拠点のみ換算をおこなう。 PL、BS 科目は換算対象とする、販売数量とKPI科目(経営指標)は換算をおこなわない。 通貨はローカル通貨のみ換算対象、JPY は換算対象外とする。 取引明細データを通貨レートを利用して乗算する。 小数点以下、4 桁から四捨五入をおこなう。 <p>(4)入出力条件</p> <ul style="list-style-type: none"> 入力条件は当年、当月度を引数とする。 取引明細データへ登録をおこなう。 	<p>国内拠点は JPY、ローカルを格納する。 海外拠点はローカルを格納する。 PL 科目、BS 科目、販売数量を格納対象とする。KPI 科目(経営指標)は格納対象外とする。</p> <p>(5)処理詳細</p> <ol style="list-style-type: none"> 共通モジュール呼び出し 本機能において利用する共通モジュールを呼び出す。 データ取得 引数から当年度、当月を取得し、取引明細データを取得する。 取得後、取引明細データは変数へ格納する。 国内拠点または海外拠点を判定する。 国内拠点は換算処理をおこなわない(5)の処理へ進む。 海外拠点は換算処理をおこなうため(4)の処理へ進む。 換算処理 計算式: 海外拠点データ(USDレート)の場合、 取引明細データ * 当月通貨レート(\$) ※計算結果は小数点以下、4 桁から四捨五入をおこなう。 換算後の取引明細データを格納する。 取引明細データへ換算後データを格納する。 終了処理 正常終了の場合、正常系のログを出力して終了する。
--	---

図3 基本仕様書の、文章表現による仕様の例

(2) 各入力条件を組み合わせたときの出力結果が明記されていない。

総じて、最も重要な欠陥は、条件と処理の組合せが網羅できているか判然としない点にある。このため、詳細設計者や総合テスト設計者が基本仕様の内容を誤解する恐れがある。

3. 提案

本研究では基本設計における入力条件と結果の正確性、およびそれらの組合せの網羅性を向上させるため、CFD 法を使ったデシジョンテーブルを取り入れることを提案する。これにあたり、この方法を検討対象とした経緯、CFD 法を使ったデシジョンテーブルの作成方法、および添付資料1の例に対するCFD法の適用について説明する。

3.1 基本仕様に対するテスト技法の導入

SQIP 研究会の分科会活動の中で、研究員は講義と演習を通じて各種のテスト技法について学習する機会を得たが、ここで学習した知識に基づいて、基本仕様の改善に取り入れる具体的なテスト技法を探す。図3はテスト技法の種類と目的の位置づけを表現している^[4]。基本仕様書は、要件定義において検討した実装すべき機能仕様を網羅的に記述することが要求される。組合せテスト技法は機能仕様の条件-結果の組合せを表現できる技法であり、機能仕様の網羅性を確認するには有効な技法である。

一方、探索的テストやエラー推測といった、疑わしい箇所をピンポイントで探し出して順次絞り込んでいく技法(図3の左半分)は、機能仕様を網羅的に表現できない。このため、これらの技法は、今回は除外とした。

さらに、図3の中から組合せテスト技法で基本仕様を正確に表すために最適な技法を絞り込む。

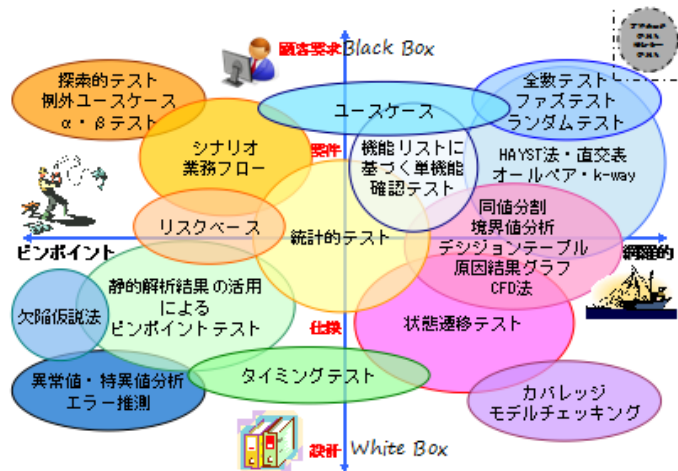


図3 テスト技法ポジショニングマップ

第5分科会（テスト技法グループ）

表1 組合せテスト技法を基本設計に取り入れるメリット／デメリットの比較

	メリット	デメリット
デシジョンテーブル	条件－処理の組合せをモデル化できる。 技法の習得が比較的容易。 レビューしやすい。	組合せが多くなるとケース数が爆発する。 組合せ漏れや誤りを防ぐ仕組みがない。
原因結果グラフ	因子間の論理関係をモデル化できる。	技法の習得に時間がかかる。 テストケースを減らす仕組みとして、有効系と無効系の区別が無い。
組合せテスト (直交表、オールペア法など)	—	論理関係の組合せ条件自体の適正化を目的としたものではないため、基本仕様の表現には使えない。
CFD法	条件－処理の関係をモデル化できる。 無駄な条件を省略できる。 技法の習得が比較的容易。 レビューしやすい。	論理和(OR)の関係が増える则有効系が増える。 論理構造が明らかでないとケースが漏れる。

デシジョンテーブル^[6]は、組合せが多くなるとケース数が爆発的に増加する。また組合せの漏れや誤りを防ぐ仕組みが弱い。原因結果グラフ^[6]は、テストケースを減らす仕組みとして、有効系と無効系の区別が無い。直交表ベースのテストやオールペア法は、論理関係の組合せ条件自体の適正化を目的としたものではない^[7]ため、基本仕様の表現には使えない。結果、これらの技法も除外とした。

CFD法は、条件と処理の組合せをデシジョンテーブルでモデル化できる上に、無駄な条件を省略してケース数を削減する仕組みを持っているため、基本仕様の改善に取り入れるテスト技法として最も適していると考えた。

3.2 CFD法について

3.2.1 CFD法とは

テスト技法としてのCFD法は、同値分割、デシジョンテーブルなど伝統的なテストケース設計技法を実践の立場で利用・体系化したものである。総合テストのテストケース数は、条件の組合せ数に応じて爆発的に増加する傾向にあるが、単体テストにおける全組合せ網羅を前提に、簡単な図と記号で仕様を分析するブラックボックステストにより、合理的にテスト項目を減らすことができる^[3]。

3.2.2 CFD法を使った基本仕様の作成

ここでは、CFD法を用いて入力条件を抽出する手順を説明する。

- (1) 条件を抽出して同値分割する。
- (2) 処理を抽出して同値分割する。
- (3) 処理は有効系（正常処理）と無効系（異常処理）に分類する。
- (4) 条件と処理の関係を原因流れ図で整理する。
- (5) 原因流れ図にある論理関係をデシジョンテーブルに展開する。

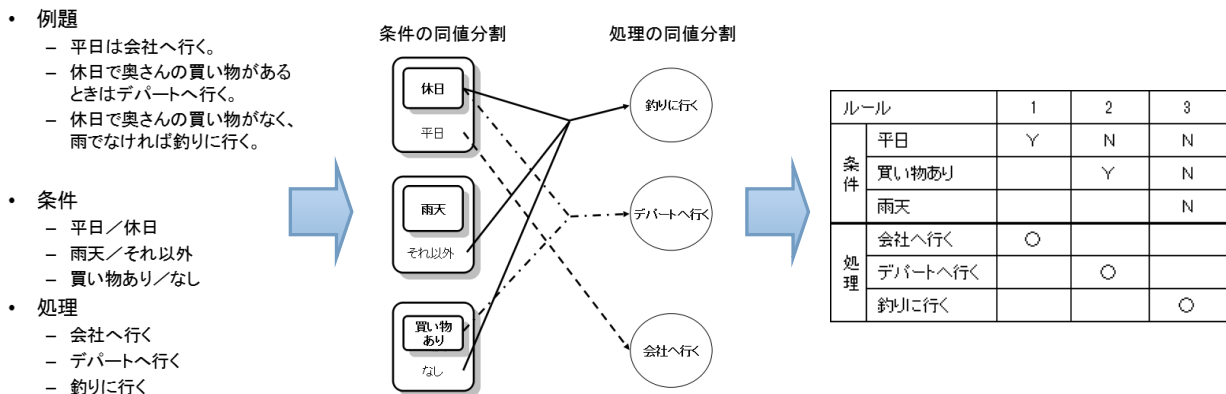


図4 CFD法を使ったデシジョンテーブルの作成例

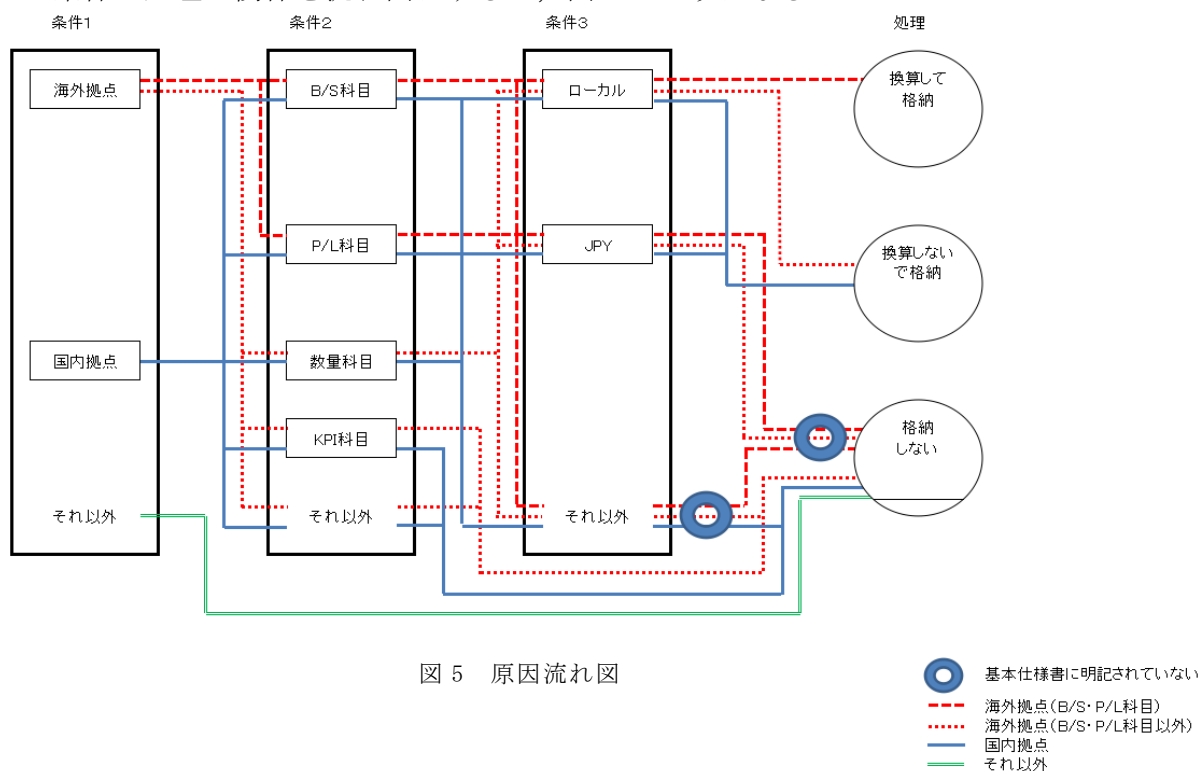
第5分科会（テスト技法グループ）

3.3 基本設計でCFD法を適用して、条件と処理の組合せを抽出する

上述2.2で取り上げた基本仕様書（添付資料1）について、3.の提案に沿ってCFD法を適用し、デシジョンテーブルを作成する。まず、次のように条件と処理について、完全同値分割を行って抽出した。

- 条件
 - 組織（国内拠点／海外拠点／それ以外）
 - 勘定科目（PL科目／BS科目／数量科目／KPI／それ以外）
 - 通貨（JPY／ローカル／それ以外）
- 処理
 - 換算（する／しない）
 - 格納（する／しない）

この条件と処理の関係の流れ図にすると、図5のようになる。



さらに、この流れ図を元にデシジョンテーブルを作成し、添付資料1の仕様にある機能要件、入出力条件、処理詳細を添付資料2のとおり整理する。

4. 評価と考察

4.1 評価に当たっての前提条件

評価の対象としたプロジェクトの特性により、勘定科目の「その他科目（KPI科目）」を有効系として扱うことにした。また、通貨の「その他通貨」（USD, EUR, CNY）についても同様に有効系とした。

ここで取り上げる基本設計は、適用する業務の論理構造を明確にすることを目的としているため、コンピュータ処理（ファイルの入出力、画面操作など）は除外する。

第5分科会（テスト技法グループ）

4.2 評価方法

4.2.1 条件と処理の組合せの比較

表2はCFD法で作成して基本仕様書に導入したデシジョンテーブルである。表2の中で、網掛けされた列は添付資料1の基本仕様書の文章で漏れている条件－処理の組合せである。この表を見ると、既存の基本仕様書は条件－処理の組合せの半分程度しか網羅できていないことが分かる。

表2 CFD法で作成したデシジョンテーブル

規則(単適合)		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
条件	組織	国内拠点	Y	Y	Y	Y	Y	Y	Y	Y	Y											
		海外拠点											Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	勘定科目	PL科目	Y	Y	Y								Y	Y	Y							
		BS科目				Y	Y	Y								Y	Y	Y				
		数量科目							Y	Y	Y								Y	Y	Y	
		その他科目										Y										Y
	通貨	JPY	Y			Y			Y				Y			Y			Y			
		ローカル		Y			Y			Y				Y			Y			Y		
		その他通貨			Y			Y			Y				Y			Y			Y	
結果	換算	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	Y	N	N	N	N	N	
	格納	Y	Y	N	Y	Y	N	Y	Y	N	N	N	Y	N	N	Y	N	N	Y	N	N	

※網掛けは既存の基本仕様書で漏れている条件－結果の組合せ

4.2.2 総合テストケースの抽出

上述3.3で作成した仕様（添付資料2）を評価するため、研究員の所属企業のテスト設計者に、既存の基本仕様書（添付資料1）および今回作成した基本仕様書（添付資料2）から総合テストのケースを抽出してもらい、成果物（添付資料3～6）を比較した。Aは入社1年目の新人、Bは入社9年目でソフトウェア開発の全フェーズを満遍なく経験している中堅社員である。なお、添付資料1からテストケースを抽出する時点では、添付資料2はテスト設計者に提示していない。

(1) 既存の基本仕様書(添付資料1)から抽出する。

抽出されたテストケース数は表3のとおりとなった。

表3 既存の基本仕様書からのテストケース抽出

テスト設計者	A	B
抽出されたケース数	16	25
仕様書のスコープ内にあるケース数	16	16
結果誤り	4	7
正しいケース数	12	9
抽出漏れ	4	4
目的外のケース数	0	9
作成工数(時間)	約3時間	約2時間

※正しいケース数は添付資料2のデシジョンテーブルの列数とする。

テストケース漏れの内容を詳しく確認したところ、テスト設計者A、Bに共通して、通貨が「その他通貨」のテストケース（表2の3、6、9、13、16、19列）が漏れていた。また、以下のテストケースについて、結果の誤りが見られた。

- A、B共に、海外の「PL科目」、「BS科目」、「KPI科目」かつ通貨が「JPY」のとき、格

第5分科会（テスト技法グループ）

納する。（表2の11, 14, 17列, 正しくは格納しない.）

- Bについて, 「国内の販売数量」, かつ「通貨がローカル」のとき, 格納しない。（表2の8列, 正しくは換算せずに格納する.）

(2) デシジョンテーブルを取り入れた基本仕様書（添付資料2）から抽出する場合のテストケース数は表4のとおりとなった.

表4 CFD法によるデシジョンテーブルを取り入れた基本仕様書からの抽出

テスト設計者	A	B
抽出されたケース数	22	24
正しいケース数	20	20
目的外のケース数	0	4
重複しているケース数	2	0
作成工数(時間)	1時間以内	1時間以内

※正しいケース数は添付資料2のデシジョンテーブルの列数とする.

テスト設計者Aによるテストケース数の差は, 「国内拠点」かつ「KPI科目」のケースが誤って重複して作成されていたために発生した. また, テスト設計者Bによるテストケース数の差は, このテスト設計の範囲外の条件（詳しくは, データ・マスタの異常なケース）を追加したものであった.

5. 結論と考察

5.1 確認できた効果

5.1.1 組合せ条件の網羅性

既存の基本仕様書（添付資料1）でテストケースを作成したところ, 正しいテストケース20件のうち, テスト設計者Aが12件, Bが9件と, 漏れや誤りが非常に多い結果となった. 一方, CFD法によるデシジョンテーブルを取り入れた基本仕様書（添付資料2）の場合, デシジョンテーブルにある条件-処理の組合せに従ってテストケースを抽出すれば良いため, 想定していたテストケースのパターンが全て存在し, テスト設計者A, Bともテスト漏れは0件であった.

5.1.2 テスト設計工数の削減

既存の基本仕様書（添付資料1）ではテスト仕様書を熟読しないとテストケースを抽出できないため, 2~3時間を要していた. 一方, CFD法によるデシジョンテーブルを取り入れた基本仕様書（添付資料2）の場合, デシジョンテーブルにある条件-処理の組合せに従ってテストケースを抽出すれば良いため, テスト設計者A, Bともに1時間以内でテストケースの抽出を完了した.

結果として, 基本設計にCFD法を取り入れたことで, 総合テスト設計時の条件-結果の組合せの漏れや誤りを防止できること, および総合テスト設計の工数が削減できることが実証できた.

5.2 その他の効果

CFD法によるデシジョンテーブルを取り入れた基本仕様書には, 次の効果もある.

- (1) 条件-処理の組合せの一意な表現により, レビューが容易になる.
- (2) 詳細設計やプログラミングにおいて, 条件分岐が整理され可読性・保守性が向上する.

5.3 実用的な工夫点

第5分科会（テスト技法グループ）

テスト技法としてのCFD法は、プログラムに展開された仕様の論理構造を整理する手順として、同値分割を元に原因流れ図を作成し、条件－処理の組合せをデシジョンテーブルで表現する（添付資料7参照）。しかし、基本設計の段階では論理構造が決まっておらず、むしろ設計者が要求仕様から論理構造を想定して、基本設計に展開しなければならない。基本設計にCFD法を取り入れて、要求仕様をどのように条件と処理に分割し、単純な論理構造にするかを考えることが必要である。この点で、本研究では、条件－処理の組合せを確実に把握するため、表2にあるような無効系を意図的に残したデシジョンテーブルを作成した。さらに無効系を整理することで、表5のようにコンパクトなデシジョンテーブルを作成することができる。

表5：コンパクトなデシジョンテーブルの例

規則(単適合)		1	2	3	4	5	6	7	8	9	#	
条件	組織	国内拠点	Y	Y	Y	Y	Y					E L S E
		海外拠点							Y	Y	Y	
	勘定科目	PL科目	Y			Y			Y			
		BS科目		Y			Y			Y		
		数量科目			Y				Y		Y	
		その他科目										
	通貨	JPY	Y	Y	Y							
		ローカル				Y	Y	Y	Y	Y	Y	
		その他通貨										
	結果	換算	N	N	N	N	N	N	Y	Y	N	
格納		Y	Y	Y	Y	Y	Y	Y	Y	Y	N	

国内拠点でPL/BS/数量の通貨がJPYまたはローカルは換算せずにデータ格納のみ
海外拠点でPL/BSでローカルだったら換算してデータ格納する
海外拠点で数量でローカルは換算せずにデータ格納する
上記以外はデータ格納しない

5.4 基本設計者によるCFD法の習得

CFD法を基本仕様に取り入れる課題の一つとして、基本設計者に対するCFD法の教育・普及がある。研究員は研究活動の中で、CFD法を使ったデシジョンテーブルの作成方法を、およそ4時間の講義と演習で学習して、今回の研究に活用できた。この程度のコストで基本設計のスキルアップを図れる点も、CFD法を取り入れるメリットである。

5. 今後の課題

組織的な実践にあたっては、基本設計者向けに適切な教育・普及のコンテンツを作成しなければならないと考えている。これは、各研究員が本研究の成果を所属企業へフィードバックする際の課題とする。

6. 引用文献

- [1] SQuBOK 策定部会, “ソフトウェア品質知識体系ガイド”, 第2版, 170, 2014
- [2] 石原一宏, “仕様整理に「デシジョンテーブル」を使ってみよう”, http://gihyo.jp/dev/serial/01/test_up/0005
- [3] 堀田文明, “CFDの考え方とデシジョンテーブル”, 30SQiP 第5分科会, 1, 2014
- [4] 秋山浩一, “高信頼化ソフトウェアのための 開発手法ガイドブックセミナー”, IPA/SEC セミナー, 4, 2011
- [5] 秋山浩一ほか, “ソフトウェアテスト HAYST 法入門”, 7, 2007
- [6] 加瀬正樹, “原因結果グラフ技法を学んでみよう！使ってみよう！”, <http://jasst.jp/archives/jasst10t/pdf/s2.pdf>
- [7] 富士ゼロックス(株), “HAYST法によるソフトウェア組合せパターン作成”, 29, 2005