欠陥混入メカニズムの知識を活用したDRBFMの提案

Proposal of DRBFM utilizing knowledge of "Defect injection mechanism"

株式会社デンソークリエイト
DENSO CREATE INC.

○柏原 一雄　新留 光治[1)]　藤田 亮太[1)]　周 廣有[1)]
○Kazuo Kashiwabara　Mitsuharu Shintome[1)]　Ryota Fujita[1)]　Guangyou Zhou[1)]
小林 展英[1)]　竹下 千晶[1)]　林 香織[1)]
Nobuhide Kobayashi[1)]　Chiaki Takeshita[1)]　Kaori Hayashi[1)]

Abstract

In derivational development, preventing "the lack of change in base software" is an important issue. In our organization, in order to solve this problem, we introduced methods such as XDDP and confirmed that it is effective. However, with respect to "parts affected by changes in design constraints", "the lack of change in base software" cannot be prevented. In this paper, in order to solve the problem, we propose DRBFM utilizing knowledge of "Defect injection mechanism". As a result of applying this method to real development, the effect was confirmed.

1. はじめに

ベースソフトを部分的にしか理解できていない状況での作業が強いられる派生開発において，変更漏れに分類される欠陥の流出を防止することは重要な課題である．我々の組織でも，派生開発において，変更漏れに分類される欠陥の割合が多い傾向があり，対策が必要であった．

派生開発において，ソースコードの変更箇所を漏れなく特定するために，XDDPをはじめとする様々な手法が提案されている．我々のプロジェクトでは，ソースコードの変更箇所の特定漏れを防止するため，XDDP，影響波及パス分析法，DRBFMを適用している．これらの手法を適用していても，「ベースソフトの設計制約の変化が影響して変更が必要となる箇所」については，変更漏れを完全に防止することはできていない．

変更箇所が特定できているときの調査のやり方を分析し，設計制約の変化による影響箇所は，DRBFMのやり方を改善することで抽出可能であると考えた．DRBFMの結果が人の知識・経験に依存することから，我々のプロジェクトではDRBFMを実施していても欠陥を防止し切ることができなかった．本研究では，「設計制約の変化から影響を受ける箇所」を対象に，変更箇所の特定漏れ防止を目的として，DRBFMで必要となる知識を明らかにしたうえで，その知識を蓄積・活用可能とすることを課題とした．

本稿では，発生し得る欠陥を予測するために，欠陥が混入するメカニズム（以降，欠陥混入メカニズムと呼ぶ）を表現する必要があるという考えを示す．更に，「設計制約の変化から影響を受ける箇所」を対象に，変更箇所の特定漏れ防止を目的として，DRBFMにおいてソフトウェアの欠陥混入メカニズムの知識を活用・蓄積する手法を提案する．提案手法は，「欠陥混入メカニズムの知識の表現方法」と「欠陥混入メカニズムの知識の蓄積方法」を技術要素とする．

提案手法の有効性を確認するために，実験では以下の評価を実施した．

- 欠陥混入メカニズムの知識を入力に，発生し得る欠陥を予測することが可能か？
- 過去に発生した変更箇所の特定漏れを防止することが可能か？
- 実開発に適用し，欠陥混入メカニズムの知識の蓄積・活用が可能か？

評価の結果，提案手法により，担当者の欠陥を予測する能力を向上させることができ，過去に発生した変更箇所の特定漏れも防止できることが確認できた．また，実開発においても，欠陥混入メカニズムの知識を蓄積・活用でき，新たな心配点を抽出できることを確認した．提案手法に

株式会社デンソークリエイト　DENSO CREATE INC.
愛知県名古屋市中区栄 3-1-1　Tel: 052-238-0460　e-mail:kashiwabara@dcinc.co.jp
3-1-1, Sakae, Naka-ku, Nagoya-shi, Aichi, Japan
1) 株式会社デンソークリエイト　DENSO CREATE INC.

【キーワード：】派生開発，影響分析，DRBFM，故障モード，障害分析，欠陥

より，派生開発における変更漏れを防止する効果が期待できる．

これ以降の本稿の構成は次のとおりである．以降，2 章で現状分析の結果と研究の課題を示す．3 章では，課題解決の参考とした先行研究を示す．4 章では，考案した解決策を提案する．5 章では，提案手法の評価結果と考察を示す．6 章では，まとめと今後の進め方を示す．

本稿では，不具合，欠陥という言葉を表 1 のように定義し，使い分ける．また，設計制約という言葉も定義する．

表 1 用語と意味

用語	意味
不具合	ソフトウェアが期待結果を果たせていない状態．欠陥により不具合が発生する．
欠陥	仕様書やソースコードなどのプロダクトに含まれる不正確な記述箇所．
設計制約	設計では，ある前提条件のもとに，ある要求を満たす構造等を考案する．設計制約は，この前提条件の一部である．仕様は期待結果を示すが，設計制約は期待結果を示さないこともある．例えば，「機能の並行実行不可」「機能のキャンセル不可」「データの参照可能条件」「バッファへの格納可能データ数」などを設計制約と位置付ける．設計制約は変えられるものである．

2. 課題設定

2.1 現状分析

ベースソフトの変更箇所特定漏れの分類を行い，それぞれの変更箇所のタイプ毎に変更漏れ変更漏れ防止する方法を対応付けた．現状，防止し切れていない変更漏れの事例から，DRBFM のやり方に問題があること明らかにした．

(1) ベースソフトの変更箇所特定漏れの分類

組織の流出不具合情報を分析し，派生開発で特定が漏れるソースコードの変更箇所には，次の 3 つのタイプがあることがわかった．

A) 変更仕様から直接特定可能な箇所[1]
変更要求を実現するために，変更が必要な関数・データ等である．変更要求から漏れなく変更仕様が抽出できていれば，変更仕様から特定可能である．ベースソフトの作りを理解していなくても特定可能である．

B) 設計制約の変化から影響を受ける箇所
変更要求に対応すると，ベースソフトで設計の前提条件としていたことが変わってしまうことや変えなければないことがある．この設計制約の変化に対応するために，変更が必要となる箇所である．ベースソフトの作りを理解していなければ特定不可能である．

C) ソースコードの変化点から直接影響を受ける箇所[3]
A)と B)で特定した変更箇所が，制御の流れとデータの流れを介して影響を与える箇所である．変更箇所が複数機能間の共通関数・共有データ等の場合に，デグレードを防ぐために，変更が必要となる箇所である．

組込みソフトウェア開発では，処理時間やリソース使用量等の要求に対応するために，制約を設けたうえで，設計をする場合が多い．そのため，B)の「設計制約の変化から影響を受ける箇所」があり，変更箇所特定が必要となる．

(2) 変更箇所のタイプ毎の変更漏れを防止する方法

A)の「変更仕様から直接特定可能な箇所」については，XDDP[2]を適用することで，変更漏れを防止することが可能である．B)の「設計制約の変化から影響を受ける箇所」については，変更要求・変更仕様から直接抽出することは困難である．そのため，DRBFM[4]で設計成果を入力として影響箇所・変更箇所を特定することが有効な手段の 1 つである．C)の「ソースコードの変化点から直接影響を受ける箇所」については，統合テストにおいて影響範囲に対するテスト漏れを防止する影響波及パス分析法[3]と回帰テストを適用することで，変更漏れを防止することが可能である．

(3) 防止し切れない変更漏れの事例

我々のプロジェクトでは，XDDP と影響波及パス分析法を活用した結果，A)と C)に分類される変更箇所については，変更漏れを防止できた．しかし，B)の「設計制約の変化から影響を受ける箇所」については，変更箇所の特定漏れが防止しきれなかった．この欠陥の事例を表 2 に示す．

表 2 「設計制約の変化から影響を受ける箇所」の特定漏れに起因する欠陥事例

変更仕様概要	機能実行中に，他機能の実行が要求されても，実行中機能の処理をキャンセルせず継続する	バッファのサイズを縮小可能とする
欠陥内容	共有データの排他処理漏れ（共有データの個別データ化漏れ）	バッファオーバーフローのガード処理漏れ（一部の箇所のみ）
欠陥の要因: ベースソフトの設計制約の変化	キャンセル不可となったことで並列実行不可能であった機能が並列実行可能となる	バッファサイズの制約が変化
欠陥の要因: ベースソフトの設計	機能間で共有しているデータあり	バッファを保持しているユニットとは別ユニットからバッファへの書き込みを実施する箇所あり

(4) DRBFM の問題点

DRBFM で，変更点から心配点とその要因を抽出した結果は，人の知識・経験に依存する．そのため，我々のプロジェクトでは，DRBFM を実施していても欠陥を防止し切ることができなかった．

我々のプロジェクトでは，DRBFM ワークシート[4]を使用し，以下の手順で DRBFM を実施している．DRBFM ワークシートには，変更点，心配点，要因，影響，対策方針，対策結果を記入している．

1. USDM で表現された変更要求と変更仕様から分析対象の変更点を選定する
2. 変更点，心配点，要因，影響を整理し，DRBFM ワークシートに記入する
3. 心配点とその要因に対して対策を検討し，DRBFM ワークシートに記入する
4. DRBFM ワークシートのレビューを実施する
5. 決定した対策の実行結果を確認し，結果を DRBFM ワークシートに記入する

変更箇所が特定できていないとき，DRBFM の結果には，以下の問題点があった．

・心配点の抽出漏れ
　設計制約の変化に伴い発生の可能性のある欠陥を予測できていない．
・心配点の要因の見逃し
　予測した欠陥は，ベースソフトの設計がどのような場合に混入するか特定できていない．
・心配点抽出や要因特定の起点となる変化点の抽出漏れ
　変更要求・変更仕様をもとに変化する設計制約を特定できていない．

多田[5]は，DRBFM において，意図したものの変更に伴う相違点を「隠れた相違点」と呼び，「隠れた相違点」を考慮に入れることが，心配点と要因を挙げるうえで重要になると述べている．また，この「隠れた相違点」の洗い出し結果には技術力の差が表れるとも述べている．心配点と要因の抽出だけでなく，変化点の特定結果にも，知識・経験の差が表れる．

設計制約の変化による影響箇所は，DRBFM を実施する技術者の知識・経験を補うことで，抽出可能であると考えた．

2.2 課題提起

本研究では，「設計制約の変化から影響を受ける箇所」を対象に，変更箇所の特定漏れ防止を目的として，DRBFM で必要となる知識を明らかにしたうえで，その知識を蓄積・活用可能とすることを課題とした．

組織で流出した変更漏れに分類される欠陥を，変更箇所の 3 つのタイプに分類し割合を算出す

ると，A)27%，B)46%，C)27%である．このことからも，B)の「設計制約の変化から影響を受ける箇所」を特定可能とする手法が開発できれば，多くの欠陥の流出防止に繋がる効果が期待できる．

3. 先行研究

課題の解決策となる手法を開発するうえで，参考とした ABC 構造という考え方を示す．また，先行研究を参考に，本研究で検討したことを示す．

(1) ABC 構造

飯塚[6] は，「ある性質・属性を有するものが，あるストレス・条件にさらされると，ある現象がおきる」という知識をもって，具体的対象に一般原則でいう現象がおきるかを考えることによって，不具合が予測できると述べている．また，トラブル現象のメカニズムを表現するときに「A という性質を持つものが，B という条件（ストレス）にさらされると，C という不具合モードが起きる」という ABC 構造を使用することを提案している．

この考えを参考にして，DRBFM の入力とする知識とするソフトウェア欠陥の混入メカニズムを表現することにする．

(2) 機能 - 心配点マトリクス

安田[7]により，変更要求仕様書（USDM）を活用し，DRFBFM を取り入れた未然防止手法が提案されている．変更要求から心配点を漏れなく抽出するために，機能と不具合事象の関係を入力として，従来の DRBFM では発見できなかった心配点や原因が抽出できた成果が報告されている．

本手法では，「機能」をキーとして心配点を抽出できるようにしている．本研究では，「設計制約の変化から影響を受ける箇所」を特定するために，「機能」ではなく「設計制約の変化」から心配点が抽出しやすくなるように，知識の表現方法を検討する．

(3) ソフトウェア欠陥予測アルゴリズム

SQiP 研究会第 7 分科会[8]により，欠陥混入メカニズムを表現･蓄積する手法が提案されている．欠陥混入メカニズムを蓄積するために，欠陥混入メカニズムのモデリングのステップを以下のように定義している．

1. 過去の不具合事例をもとに，欠陥混入の主要因の関連をモデリングする．
2. 業務や製品固有の言葉を避け，一般化して表現する．
3. 欠陥混入メカニズムを理解容易性・納得性・利用可能性の観点でレビューする．

本手法を参考として，欠陥混入メカニズムを蓄積する手順を検討した．ただし，過去の不具合事例をもとに欠陥混入メカニズムのモデリングをする場合，次の原因により効率的に多数の知識を集めることが難しい．本研究では，この問題を解決可能な知識の蓄積方法を検討する．

- 必要となる情報（欠陥とその要因の関係）が不足していることが多い．
- 不具合の発生数が少ない場合には，十分な知識を集めることができない．

4. 解決策の提案

「設計制約の変化から影響を受ける箇所」を対象に，変更箇所の特定漏れ防止を目的として，DRBFM においてソフトウェアの欠陥混入メカニズムの知識を蓄積・活用する手法を提案する．提案手法は，「欠陥混入メカニズムの知識の表現方法」と「欠陥混入メカニズムの知識の蓄積方法」を技術要素とする．

4.1 課題の解決方針

(1) 欠陥混入メカニズムの知識の表現方法

欠陥混入メカニズムは，ABC 構造を参考に，「A という設計のベースソフトが，B という制約の変化にさらされると，影響箇所の特定漏れにより，C という欠陥が起きる」という表現とした．つまり，図 1 に示すような「ベースソフトの設計制約の変化」と「ベースソフトの設計」と「欠陥」の関係を表現する．

この知識を活用することにより，提案手法では，「ベースソフトの設計制約の変化」から発生し得る「欠陥」を予測しやすくする．更に，「欠陥」の見逃しを誘発させるような「ベースソフトの設計」の情報も併せて示すことで，ベースソフトの調査時に注意すべき箇所が明らかにし，影響箇所の見逃しを防止しやすくする．

図 2 に欠陥混入メカニズムの知識の DRBFM における活用タイミングを示す．DRBFM における「設計制約の変化を特定」「発生し得る欠陥を予測」「影響箇所を抽出」という作業で，欠陥混入メカニズムの知識を活用する．

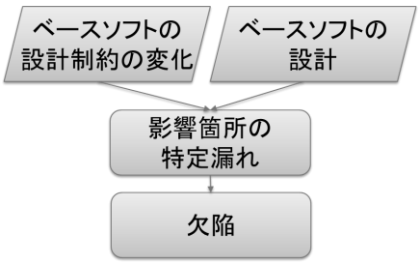

図 1 欠陥混入メカニズムの構造

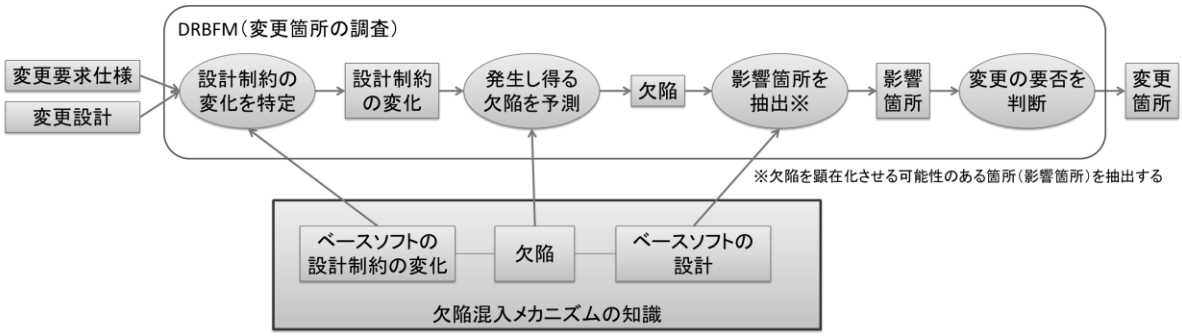

図 2 欠陥混入メカニズムの知識の活用タイミング

(2) 欠陥混入メカニズムの知識の蓄積方法

欠陥混入メカニズムは，図 3 に示すように，DRBFM ワークシートを入力に，蓄積する方針とした．DRBFM ワークシートには，心配点とその要因が必ず示されているため，効率的に欠陥混入メカニズムの知識を表現することが可能となる．また，DRBFM ワークシートには，流出した欠陥だけではなく流出を防止できた欠陥も示されるため，多数の知見を集めることが可能となる．

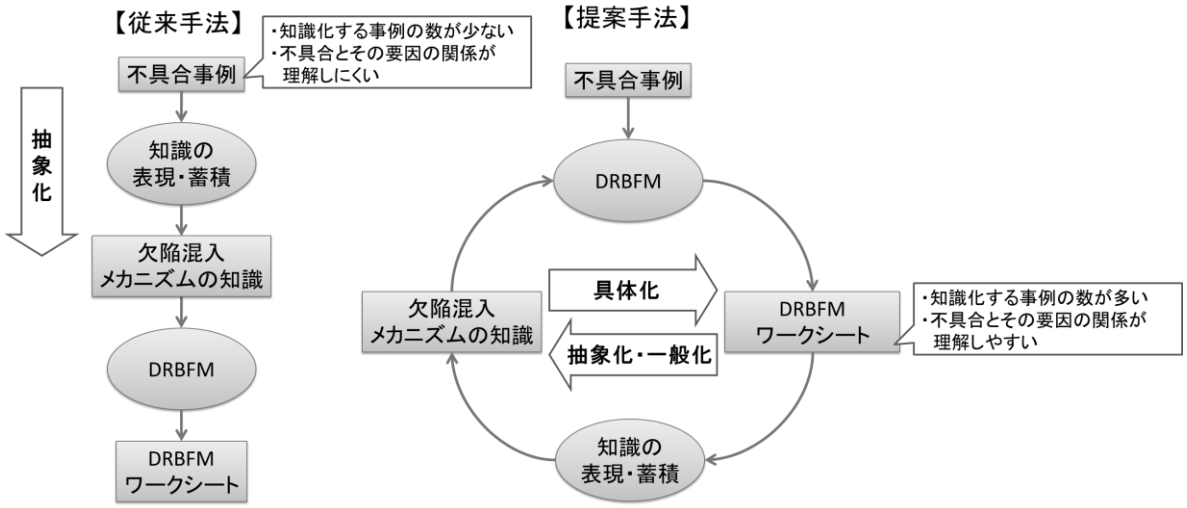

図 3 欠陥混入メカニズムの知識の蓄積方法

4.2 欠陥混入メカニズムの知識の表現方法の定義

欠陥混入メカニズムの知識は，以下に示す 3 つの要素で表現する．

・設計制約の変化
ベースソフトの設計制約の変化を示す．欠陥混入メカニズムの知識を検索するためのキーとなる情報である．

・欠陥

設計制約の変化により，引き起こされる可能性のある欠陥を示す．ソースコードの欠陥を表現する．

・ベースソフトの設計

欠陥の要因となるベースソフトの設計を示す．予想した欠陥が発生し得る箇所を絞り込むために利用する知識である．変更箇所特定時に注目すべき点を示す．例えば，「複数機能からアクセスされる外部変数」「保持されているユニットとは別ユニットから書き込みがされる外部変数」等を示す．

欠陥混入メカニズムの知識を蓄積した成果物を，DIM 辞書（Defect injection mechanism 辞書）と呼ぶ．DIM 辞書には， DRBFM で参照する順に左から情報を並べる．図 4 に DIM 辞書のイメージのイメージを示す．

No	設計制約の変化	欠陥	ベースソフトの設計
1	機能が並列実行不可能から変更実行可能に変化	・共有データの排他処理漏れ	・機能間で共有しているデータ(外部変数)あり
2	処理のキャンセル不可能からキャンセル可能に変化	・デッドロックの対策処理漏れ	・処理実行中の状態を複数のレイヤ(ユニット)でそれぞれ管理
3	処理が異常終了することなしから異常終了することありに変化	・デッドロックの対策処理漏れ	・処理実行中の状態を複数のレイヤ(ユニット)でそれぞれ管理
4	同期処理が非同期処理に変化（処理中に他の要求を受け付けられるようになる）	・キューあふれ	・処理中に受け付けた別の要求は、キューに格納し、実行中処理が完了後に実行される
5	処理時間が長くなる（処理中に受け付ける要求が増える可能性がある）	・キューあふれ	・処理中に受け付けた別の要求は、キューに格納し、実行中処理が完了後に実行される
6	バッファサイズの制約が変化（バッファオーバーランが発生する可能性がある）	・バッファオーバーランのガード処理漏れ	・同じ種類のデータが格納されるバッファ(配列)が複数あり ・バッファ(配列)が保持されているユニットとは別ユニットから書き込みがされる箇所あり
7	バッファのデータの歯抜けなしから歯抜けありに変化	・データ取得時の無効値判定処理漏れ／有効値範囲判定処理誤り ・ソート処理の漏れ ・有効データ位置情報の更新漏れ	・バッファのデータの並べ替え処理なし ・バッファ(配列)のデータ参照箇所が複数 ・バッファ(配列)のデータ更新箇所が複数
8	バッファのデータの並び順に関する制約が変化	・データ検索処理誤り	・バッファのデータの並べ替え処理なし ・バッファ(配列)のデータ参照箇所が複数
9	データの有効区間が変化	・データ取得時の無効値判定処理漏れ	・データの参照箇所が複数
10	データの種類が増える（サポート対象外だったものがサポート対象となる）	・データの種類毎に処理を切り替えるべきところを、同一の処理を実行	・データの種類を意識して処理を切り替える箇所が複数

図 4 DIM 辞書のイメージ

4.3 欠陥混入メカニズムの知識の蓄積方法の定義

DIM 辞書を拡充するために，DRBFM の結果をもとに DIM 辞書に知識を蓄積する手順を定義する．

1. DRBFM の実施
   過去の不具合事例，レビュー指摘などを入力に，DRBFM を実施する．DIM 辞書が存在していれば，DIM 辞書も入力として DRBFM を実施する．
2. DRBFM ワークシートからの欠陥混入メカニズム知識の抽出
   1.で作成した DRBFM ワークシートを入力に,設計制約の変化に起因する心配点を抽出する．
3. 欠陥混入メカニズム知識の一般化
   2.で抽出した心配点とその要因を，「A という設計のベースソフトが，B という制約の変化にさらされると，影響箇所の特定漏れにより，C という欠陥が起きる」関係に当てはめ，一般化して表現する．
4. 欠陥混入メカニズム知識の DIM 辞書への登録
   3.で表現した欠陥混入メカニズムの知識を DIM 辞書に登録する．同じ知識が複数存在している状態ならないように，表現を見直す．必要に応じて既存の知識の表現も見直す．
5. 欠陥混入メカニズム知識のレビュー
   DIM 辞書の内容を，理解容易性・納得性・利用可能性の観点でレビューする．1.の DRBFM で DIM 辞書を利用している場合は，既存の知識に対しても，理解しにくかった表現や不足している情報を明らかにする．
6. DIM 辞書の修正
   5.のレビュー指摘をもとに，DIM 辞書の内容を修正する．

5. 解決策の評価

5.1 評価方法

提案手法の有効性を確認するために， 3 つの評価を実施する．評価観点毎に評価方法を示す．各評価では，図 4 に示した DIM 辞書を使用する．

a. 欠陥混入メカニズムの知識を入力に，発生し得る欠陥を予測することが可能か？
   開発経験の異なる複数の技術者が，変更要求・変更仕様と DIM 辞書を入力に，変化点抽出と心配点抽出を実施する．実験では制限時間（30 分）を設ける．
   評価の前提条件は，以下に示す．

・入力となる情報：変更仕様 7 件，DIM 辞書の登録知識件数 10 件
・担当者の保有知識：入力となる変更要求・変更仕様は理解している

b. 過去に発生した変更箇所の特定漏れを防止することが可能か？
DIM 辞書を活用した DRBFM により，ベースソフト調査の再実施を行い，変更箇所が特定できるかを確認する．表 2 に示した過去に変更箇所の特定漏れが発生した 2 件の事例を対象とする．

c. 実開発に手法を適用し，欠陥混入メカニズムの知識の蓄積・活用が可能か？
実開発で作成された DRBFM ワークシートを入力に，DIM 辞書が作成可能かを確認する．
また，作成した DIM 辞書を入力に DRBFM を実施し，心配点が抽出可能かを確認する．
DIM 辞書を作成したソフトウェアとは設計が異なる別のソフトウェアの開発で DIM 辞書を活用する． DIM 辞書を使用せず DRBFM を実施した後に，DIM 辞書を入力にして，追加で心配点抽出を行う．
評価の前提となる実開発の情報を以下に示す．
・入力とした過去の DRBFM 結果の量：案件数 8 件，心配点数 合計 47 件
・DIM 辞書活用案件の開発規模：変更仕様 72 件，変更行数 約 5000LOC

5.2 評価結果

各評価観点に対して，評価結果を示す．

a. 欠陥混入メカニズムの知識を入力に，発生し得る欠陥を予測することが可能か？
表 3 に，技術者毎に抽出できた変化点数と心配点数を示す．また，開発経験の情報として，対象ソフトの開発経験年数と DIM 辞書に登録された欠陥混入メカニズムに対する経験の有無（具体的な事例が思いつくか）を示す．
全員がなんらかの変化点と心配点を抽出できた．抽出できた心配点は，技術者毎に異なる．開発経験が増えれば，抽出できる変化点と心配点も多くなる傾向がある．

表 3 欠陥混入メカニズムの知識を入力にした心配点抽出実験結果

技術者	対象ソフトの開発経験	経験済みのメカニズムの数	抽出した変化点の数	抽出した心配点の数
A	5 年以上	10 件	10 件	10 件
B	5 年以上	9 件	6 件	6 件
D	3 年以上	10 件	6 件	7 件
E	3 年以上	5 件	5 件	4 件
F	1 年未満	2 件	3 件	3 件
G	1 年未満	0 件	3 件	1 件

b. 過去に発生した変更箇所の特定漏れを防止することが可能か？
DRBFM 結果をもとに，ベースソフトの調査を再実施した結果，2 件の事例共に，「設計制約の変化から影響を受ける箇所」を特定できた．

c. 実開発に手法を適用し，欠陥混入メカニズムの知識の蓄積・活用が可能か？
実開発で作成された DRBFM ワークシートを入力に，10 件の欠陥混入メカニズムが DIM 辞書に登録できた．また，作成した DIM 辞書を入力に DRBFM を実施し，5 件の心配点を抽出できた．開発案件で抽出した心配点は 56 件あり，全体の約 10%が DIM 辞書を入力に追加で抽出できた心配点である．

5.3 結果の考察

「設計制約の変化」と「欠陥」の関係が表現された知識を入力とすることにより，DRBFM で担当者の知識を補い，発生し得る欠陥の抽出漏れを防ぐことができる．「欠陥」と「ベースソフトの設計」の表現された知識を入力とすることにより，ベースソフトの調査で注意して確認すべき箇

所が明確になり，影響箇所の抽出漏れを防止することができる．

不具合事例ではなく DRBFM ワークシートを入力にすることで，欠陥混入メカニズムの知識を確実に蓄積できる．DIM 辞書の内容をレビューすることにより，理解しやすく・利用しやすい知識となり，実開発でも新たな心配点に気づくことができた．更に，DRBFM と DIM 辞書の更新を繰り返すことで，欠陥混入メカニズムを具体化することと抽象化することが繰り返され，欠陥混入メカニズムに対する理解が深まることが分かった．

ただし，提案手法では，担当者の知識を補うことはできても，担当者の知識・経験に依存しない DRBFM を可能とすることはできなかった．DIM 辞書に示されている欠陥を経験していない技術者の場合は，知識を利用することができない．提案手法による効果を高めるには，DIM 辞書に示されている欠陥を経験していることが必要であることがわかった．

また，提案手法で効果を上げるには，そもそもベースソフトの設計制約を把握できる状態にする必要があることがわかった．設計制約が明文化されていれば，設計制約の変化も捉えやすい．

6. おわりに

派生開発において，変更要求に対応することで設計制約が変化することがある．この設計制約の変化から影響を受ける箇所で発生する変更漏れが，防止できていないことがわかった．本研究では，この問題を解決するための手法として，欠陥混入メカニズムの知識を活用した DRBFM の手法を開発した．本手法の技術要素として，「欠陥混入メカニズムの知識の表現方法」と「欠陥混入メカニズムの知識の蓄積方法」の 2 つを定義した．

この提案手法を適用することで，「設計制約の変化から影響を受ける箇所」を対象に，変更箇所の特定漏れ防止する効果があることを確認した．XDDP と影響波及パス分析法に加えて本手法を実開発に適用することで，派生開発における変更漏れ確実に防止する効果が期待できる．

今後，以下の取り組みを行い，提案手法を適用するによる効果を広げ，高める．

- 異なる複数のプロジェクトにおいて提案手法を導入し，手法を評価・改善する．
- DIM 辞書に示されている欠陥を経験し理解するための教育の実施を検討する．
- ベースソフトの設計制約を確実に把握するためにプロセスを改善する．
- HAZOP[9]等の他手法との比較評価を実施し，手法を組み合せることも検討する．
- 欠陥混入メカニズムの知識を活用し，影響箇所の特定を自動化する方法を検討する．

謝辞

本研究に対して有益なご助言を戴いた 株式会社日立ハイテクノロジーズ 飯泉紀子氏，株式会社デンソー 足立久美氏，株式会社システムクリエイツ 清水吉男氏，株式会社デンソークリエイト 山路厚氏に感謝の意を表する．

参考文献

[1]清水吉男，「派生開発における 母体に由来するバグとその対応」，JaSST'09 Tokyo，2009
[2]清水吉男，「「派生開発」を成功させるプロセス改善の技術と極意」，技術評論社，2007
[3]柏原一雄，「統合テストにおいて影響範囲に対するテスト漏れを防止する「影響波及パス分析法」の提案」，ソフトウェア品質シンポジウム 2017，2017
[4]吉村達彦，「トヨタ式未然防止手法・GD³」，日科技連出版社，2008
[5]多田直弘，「モノづくりにおける実践の DRBFM」，友月書房，2015
[6]飯塚悦功，「構造化知識工学ことはじめ」，品質 35(1) ，6-10，2005
[7]安田隆司，「変更要求仕様書を活用した未然防止方法の提案-XDDP への DRBFM 導入とその効果-」，ソフトウェア品質シンポジウム 2011，2011
[8]2014 年度 SQiP 研究会第 7 分科会，「ソフトウェア欠陥予測アルゴリズム」，ソフトウェア品質シンポジウム 2015，2015
[9]河野哲也，「ソフトウェア要求仕様における HAZOP を応用したリスク項目設計法」，ソフトウェアテストシンポジウム 2012，2012