

ソフトウェア欠陥の共起性を利用した欠陥推定手法の提案

～ 共起欠陥推定アプローチによる潜在欠陥の捕捉 ～

A Proposal of defect prediction method

using the co-occurrence of the software defects

- Capture latent defects by co-occurrence defect prediction approach -

主査 : 細川 宣啓 (日本アイ・ビー・エム株式会社)
副主査 : 永田 敦 (ソニー株式会社)
研究員 : 澁谷 将行 (株式会社トーセイシステムズ)
山崎 真一 (富士ゼロックス株式会社)

研究概要

多くの企業では、ソフトウェア開発の過程で欠陥が混入しないように予防策を講じている。しかし、欠陥は人間の誤りにより発生するため混入をなくすことは容易ではなく、市場に流出しているのが実情である。そこで我々は、欠陥は混入するという前提の上で欠陥混入のメカニズムに着目し、欠陥検出時に得られる情報を用いて他の潜在欠陥を検出する手法の確立を目指した。

本研究では、欠陥混入のメカニズムをもとに、欠陥は共起することを示し、欠陥の共起性を用いた潜在欠陥の推定を行う「共起欠陥推定アプローチ」を定義する。このアプローチにより、欠陥検出時の情報を用いて、より潜在している可能性が高い欠陥を推定する手法を提案する。

Abstract

Many companies take preventive measures to prevent contamination of defects during software development. However, since defects originate from human errors, it is not easy to eliminate contamination, and it is actually the case that they leak out to the market. Therefore, we aimed at establishing a method to detect other latent defects by using the information obtained at the time of defect detection based on the premise that defects are injected, to focus on the mechanism of defects contamination.

We define "co-occurrence defect prediction approach" which shows co-occurrence of defects and estimates latent defects using defect co-occurrence based on the mechanism of defect contamination. With this approach, we propose a method to prediction defects that are more likely to be latent using frequency of occurrence at the time of defect detection.

1. はじめに

1.1 背景と課題

ソフトウェア開発の過程で、多くの企業はソフトウェア欠陥(以降、欠陥)が混入しないように予防策を講じている。しかし、欠陥は人間の誤りにより混入するとされており、欠陥の混入をなくすことは容易ではない。欠陥が混入した工程で全ての欠陥を検出および除去することができずに後工程や市場に流出しているのが実情である。日々欠陥の検出と除去を繰り返す中で、検出した欠陥に対して、「以前に修正した欠陥と似ている」と感じる事が往々にしてある。このようなことが起こるのは、以下の理由により、検出しきれずに潜在する欠陥があるためと考えられる。

第7分科会 (Team KuKuRu)

- (1) 欠陥の検出時に、似たような欠陥がないかを調査することはよく行われる。しかし、多くの場合は欠陥という現象の除去に留まり、欠陥の混入原因を除去していない。
- (2) 似たような欠陥が潜在している可能性に気付いているが、一網打尽に対応する方法が分からないため、個別にしか検出できない。また、類似性の正体を正確に定義することが困難なため、検出しようとしても主観に頼らざるを得ない。

1.2 研究の目的

一般的に欠陥検出工程が後工程であるほど、修正コストが高くなることは広く知られている。コストの低減のためには、計画段階で欠陥混入に対策し、混入を予防することが効果的である。一方で、プロジェクトの状況は絶えず変化するものである。計画時点で全ての変化を予測することは困難であるため、計画時の対策だけでは混入を抑えられない。これらの状況から筆者らは、開発工程内で欠陥を検出した際に、検出した欠陥の生きた情報を利用して、対策を講じることができないかと考えた。

本研究では、上記の課題を解決するために、欠陥の混入原因をもとに潜在欠陥を推定する手法を提案する。また提案手法により、関連する欠陥を一網打尽で除去し、後工程への欠陥流出抑制を行える可能性を明らかにする。

提案手法の実現性を証明するために、欠陥には同一原因を有して異なる欠陥が存在する性質（以降、欠陥の共起性）があると仮定し、以下のRQを設定した。

RQ1：欠陥にはどの程度共起性があるのか？

RQ2：欠陥の共起性を利用した、普遍的な潜在欠陥の見つけ方はあるのか？

2. 先行研究

2つの観点から先行研究の調査を行った。1つ目の観点として、本研究では欠陥が作り込まれる原因に着目し、複数の欠陥情報の関連性を取り扱う必要がある。そこで、以下の2件の研究をベースとして活用できると考えた。

(1) 欠陥モデリング

欠陥が作り込まれるメカニズムを表現する手法として、Project Fabre^[1]の欠陥モデリングがある。この手法では、過失（人間の判断の思考や判断の誤り）に着目し、欠陥情報を構造化して表現する手法（以降、欠陥モデル）を提案している。

(2) 欠陥予測 DB

欠陥モデルを活用して、プロジェクトの状況を示す要因をもとに、発生する可能性のある欠陥を予測する研究^[2]がある。この研究では、多数の欠陥モデルを欠陥予測DBに蓄積して欠陥予測を行う方法が提案されている。

2つ目の観点として、本研究では、欠陥を検出した時点の情報からその他の潜在欠陥を推定することを狙いとしている。同様の狙いをもつ研究として、ODC分析による事例研究^[3]がある。この研究では、検出した時点で確定する欠陥の属性を用いて欠陥分析を行い、潜在欠陥の推測を行う取り組みが示されている。しかしODC分析では、欠陥が存在する傾向/場所は示唆されるが、具体的にどのような欠陥があるかは分からない。また、汎用的であるがゆえに、特定技術に依存する欠陥や、業務ドメインに特化した欠陥は見つかりにくい傾向にある。これは一般的な欠陥の検出には有効だが、特定の重大問題を引き起こす欠陥ノウハウが蓄積されにくい性質を示唆する。

3. 提案

本稿では、欠陥の共起性を利用した潜在欠陥の推定アプローチを提案する。なお本アプローチでは、推定方法としてグラフ理論を用いる。グラフ理論は、ノード(点)とエッジ(線)をもつネットワーク構造で表現できる問題の解析に応用されている数学の理論である。

例えば、単語の共起関係や論文の共著関係などに2部グラフの構造が見られるとされる^[5]。欠陥モデルをグラフ理論に当てはめた場合、各因子（誘発因子、過失因子、欠陥）の間で繋がりを持ち、同じ因子内では繋がりをもたない3部グラフと捉えることができる。このため、グラフ理論を適用できると考えた。

3.1 欠陥の共起性

本稿では複数の欠陥モデルを統合して利用する場合、次の定義が成り立つと仮定する。

【定義】(欠陥モデル)

欠陥モデル X は、誘発因子の集合 A 、過失因子 b 、欠陥 c を用いて $X = (A, b, c)$ で定義され、グラフ $G(X) = (N, E)$ を表す。ここで、ノード集合 N 、エッジ集合 E は次で与えられる。

$$\text{定義式 1) } N = A \cup \{b, c\}, E = \{(a, b) | a \in A\} \cup \{(b, c)\}$$

【定義】(欠陥の共起性)

欠陥 c_1 と c_2 に共起性があることを次で定義する。

定義式 2) 欠陥モデル X_1, X_2 が存在し $G(X_1) = (N_1, E_1), G(X_2) = (N_2, E_2)$ のとき、 $N_1 \cap N_2 \neq \emptyset$ 。(つまり $X_1 = (A_1, b_1, c_1), X_2 = (A_2, b_2, c_2)$ ならば $A_1 \cap A_2$ または $b_1 = b_2$ または $c_1 = c_2$)

この定義により、既存の単一のバグ票によって欠陥を管理していたレベルから、本稿で提案する共起欠陥を扱うなど複雑かつ高度な推定が可能となると筆者らは考えた。

以下に具体的な例を用いて、定義の意味を示す。図1と図2の欠陥モデルはそれぞれ、筆者らが所属する組織において実際に発生した欠陥をモデリングした、単一の欠陥モデルである。ここで、例えば図1の欠陥モデル X_1 の要素 A_1, b_1, c_1 はそれぞれ下記である。

- ・誘発因子の集合 $A_1 = \{\text{機械制御のノウハウ未継承, 参考資料なし, ハーネスでの動作確認}\}$
- ・過失因子 $b_1 = \text{機械は常に同じ動作をするという思い込み}$
- ・欠陥 $c_1 = \text{物理法則を考慮していない設計}$

この場合、従来は2つの欠陥は過失因子、欠陥、表出現象が異なる関連性のない欠陥であると認識していた。しかし、欠陥の共起性の定義に従うことで、「機械制御のノウハウ未継承」と「ハーネスでの動作確認」という2つの共通する誘発因子によって混入した「物理法則を考慮していない設計」と「機械の性能以上の動作指示」という2つの欠陥には、共起性があることが示される。



図 1. 欠陥モデル 1

図 2. 欠陥モデル 2

上記の例のように、共起性を用いることで、単一で欠陥を扱ってはいは検知できなかった異なる欠陥が検知可能となる。このような、一方の欠陥をもとに検知できる他方の欠陥を、本稿では「共起欠陥」と呼称する。

3.2 共起欠陥推定アプローチ

3.1章で定義した欠陥の共起性を利用したアプローチを、我々はCOPアプローチ(Co-occurrence defect Prediction アプローチ：共起欠陥推定アプローチ)と名付けた。

COPアプローチではまず、過去に蓄積した単一の欠陥をモデル化した欠陥モデルを複数統合して、推定用欠陥モデル(グラフ)を生成する。次に、生成したグラフと現在検出された単一の欠陥をリアルタイムで照合する。その際にグラフを辿ることで、欠陥を検出した

第7分科会 (Team KuKuRu)

時点で潜在している共起欠陥を推定する。推定手順は次の2ステップからなる。

Step1. 推定用欠陥モデルの生成：

欠陥モデルの集合 D があるとき、各欠陥モデル $X \in D$ は部分グラフ $G(X) = (N, E)$ を表すため、 D から有向グラフ $G(D) = (N^*, E^*)$ を構成できる。ただしノード集合 N^* 、エッジ集合 E^* は以下で与えられる。

$$N^* = \{\text{node} | \text{node} \in N, G(X) = (N, E), X \in D\}, E^* = \{\text{edge} | \text{edge} \in E, G(X) = (N, E), X \in D\}$$

Step2. 共起欠陥の推定：

選択された欠陥 d に対する共起欠陥は、グラフ $G(D)$ において、欠陥 d と連結な欠陥として求められる。

COPアプローチを適用する場合、従来の潜在欠陥予測に比べて以下のメリットがある。

(1) 潜在欠陥の欠陥種類の推定が可能。

レイリーモデル^[4]を用いた潜在欠陥予測などでは、推定対象は潜在欠陥数である。一方、COPアプローチでは欠陥情報をもとにした推定を行うため、具体的な欠陥の種類^[4]の推定が可能になる。これにより、具体的な潜在確認や対策が可能になる。

(2) リアルタイムな推定が可能。

欠陥検出時に、検出した欠陥情報をもとに推定を行うため、計画段階では推定できなかった欠陥の抽出が可能となる。また蓄積された欠陥情報を用いるため、常に最新のプロジェクト状況を考慮した推定が可能になる。

(3) 属人性が少ない分析が可能。

グラフを用いた分析を行うことで、分析者の属人的な経験に依存する部分が少なくなる。これにより、初心者でも客観的な分析結果を得ることが可能になる。

(4) 欠陥混入の全体像を可視化可能。

一般的に品質改善技法は個別の問題に対する解決手法を提案する場合があるが、共起欠陥という一般的には知られていない概念を用いて、複雑な欠陥発生が絡み合う品質の全体像を把握することができるようになる。これにより、最初のわずかな欠陥を検出する労力で、関連欠陥を一網打尽にすることが可能になる。

4. 実験

3章で提案した内容の有効性を検証するため、以下の実験を行った。

4.1 実験1

<実験手順>

- ① 研究員が所属する組織にて発生した過去プロジェクトの欠陥情報を、BTS (Bug Tracking System: バグ管理システム) から収集する。
- ② 手順①の欠陥情報から、欠陥モデリングの定義に従い、誘発因子、過失因子、欠陥を収集し、欠陥予測DBの形式に従って登録する。
- ③ 欠陥予測DBから3.2章のStep1に基づき、推定用欠陥モデルのグラフを作成する。
- ④ 3.2章のStep2に基づき、以下のアルゴリズムを用いて全共起欠陥を求める。

【共起推定アルゴリズム】

- (1) 共起欠陥の推定を行う欠陥を選択する。
- (2) (1)の欠陥ノードと連結している過失因子ノードを抽出する。
- (3) (2)で抽出した過失因子ノードと連結している誘発因子ノードを抽出する。
- (4) (3)で抽出した誘発因子ノードと連結している過失因子ノードを抽出する。
- (5) (4)で抽出した過失因子ノードと連結している欠陥ノードを全て抽出する。
- (6) (5)で見つけた欠陥に対して、(2)～(6)を繰り返す。
- (7) 全ての連結するノードに対してチェックが完了したら終了とする。

第7分科会 (Team KuKuRu)

なお全ての実験で、フリーのグラフ作成ソフトである「GraphViz^[6]」を使用してグラフを作成する。また欠陥予測 DB からのグラフ作成や絞り込みについても、GraphViz に対応するスクリプト(.dot 形式)への変換プログラムを作成し、使用している。これはツールやプログラムを利用することで、実験の再現性を保証するためである。

4.1.1 実験1の例

実験手順①：紙面の都合上割愛する。

実験手順②：5件の欠陥モデルを、図3のように欠陥予測DBに登録する。欠陥予測DBにおいて、1件の欠陥モデルは、同一行の誘発因子、過失因子、欠陥の組で表される。この例では、1件の重複を含む5件を選択した。

欠陥ID	材料		誘発因子		方法		人		他
	入力	ベース	発生原因	発生状況	手順	プロセス	設計	製造	
1	リソース不足	OSを認識した日付フォーマット変換を行っていない	経験不足	機械は指示通りに動作するという思い込み	経験不足	OSを認識した日付フォーマット変換を行っていない	経験不足	OSを認識した日付フォーマット変換を行っていない	OSを認識した日付フォーマット変換を行っていない
2	OS、購入ソフト(外部要因)	OSを認識した日付フォーマット変換を行っていない	経験不足	機械は指示通りに動作するという思い込み	経験不足	OSを認識した日付フォーマット変換を行っていない	経験不足	OSを認識した日付フォーマット変換を行っていない	OSを認識した日付フォーマット変換を行っていない
3	OS、購入ソフト(外部要因)	OSを認識した日付フォーマット変換を行っていない	経験不足	機械は指示通りに動作するという思い込み	経験不足	OSを認識した日付フォーマット変換を行っていない	経験不足	OSを認識した日付フォーマット変換を行っていない	OSを認識した日付フォーマット変換を行っていない
4	OS、購入ソフト(外部要因)	OSを認識した日付フォーマット変換を行っていない	経験不足	機械は指示通りに動作するという思い込み	経験不足	OSを認識した日付フォーマット変換を行っていない	経験不足	OSを認識した日付フォーマット変換を行っていない	OSを認識した日付フォーマット変換を行っていない
5	OS、購入ソフト(外部要因)	OSを認識した日付フォーマット変換を行っていない	経験不足	機械は指示通りに動作するという思い込み	経験不足	OSを認識した日付フォーマット変換を行っていない	経験不足	OSを認識した日付フォーマット変換を行っていない	OSを認識した日付フォーマット変換を行っていない

図3. 欠陥情報の登録

実験手順③：誘発因子、過失因子、欠陥をグラフのノード、因子間や因子-欠陥間の関係をグラフのエッジとして、有向グラフの表現に変換する。

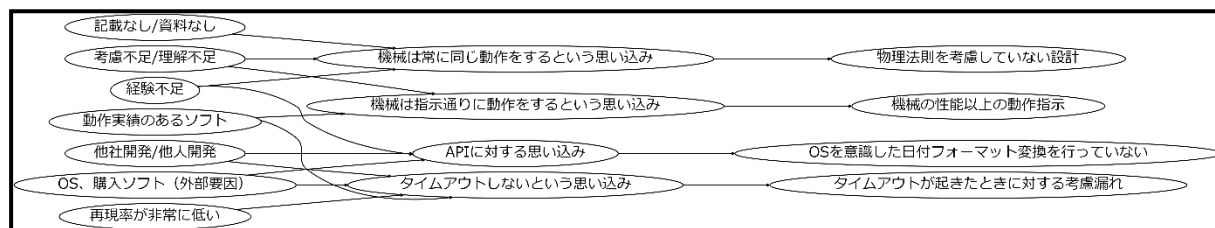


図4. 有向グラフ表現に変換

実験手順④：図4のグラフに対して共起推定アルゴリズムを適用することで、他に潜在している可能性のある欠陥を定性的に辿ることが可能となる。例えば欠陥「OSを認識した日付フォーマット変換を行っていない」を対象とすると、4件全てが抽出される。

4.1.2 実験1結果

欠陥情報41件を登録した欠陥予測DBを用いて実験した結果、付録5の有向グラフが得られた。この有向グラフから、1つの欠陥の原因である誘発因子が、異なる欠陥の原因ともなることが確認できた。この結果、欠陥には共起性があり、共起性を利用した定性的な潜在欠陥の推定が可能であることを実証した。その一方で、定性的な推定だけでは、ほぼ全ての欠陥が共起欠陥と推定され、現実的には運用が困難であるという問題点が見えた。

4.2 追加施策

実験1での問題点を解決するためには、優先して確認すべき欠陥を定量的な情報を用いて絞り込む必要があると判断した。このため、以下の追加施策を実施し実験を行った。

追加実験では、有向グラフの各エッジに、各因子の出現回数と、因子間の組み合わせ出現回数をもとにした関連強さを示す重みを付与する。付与した重みが大きいほど関連が

第7分科会 (Team KuKuRu)

強い(=共起性が強い)と捉え、検出欠陥の重み以上の共起欠陥を抽出対象とした。この方法により、共起欠陥の中でも検出欠陥よりも潜在する可能性の高い欠陥だけを定量的に絞り込めることを検証する。なお、重みに出現回数を使用するのは、以下の理由である。

- ・ 組み合わせの出現回数が多いものほど、因子間の関連性が強いと考えた。
- ・ 欠陥データの蓄積に伴い、過去の状況を考慮した重みの変化を可能とする。

4.3 実験2

<実験手順>

① 実験1で作成した有向グラフに対して、誘発因子、過失因子、欠陥の関連に以下の方法で算出した重みを付加した重み付き有向グラフを作成する。

【重みの算出方法】

(1)各誘発因子と過失因子のエッジに対して、欠陥予測DBをもとに以下の式を用いて重みを付加する。

$$\text{計算式 1) 誘発因子-過失因子エッジの重み} = \frac{INC}{TI}$$

TI : 対象誘発因子の出現回数の合計(対象誘発因子の○の数)

INC : 誘発因子と過失因子のつながりが発生した回数

(2)各過失因子と欠陥のエッジに対して、以下の式を用いて重みを付加する。

$$\text{計算式 2) 過失因子-欠陥エッジの重み} = ES * \frac{NDC}{TN}$$

TN : 対象過失因子の出現回数の合計

ES : 対象過失因子に連結する、誘発因子-過失因子エッジの重みの合計

NDC : 欠陥予測DBで過失因子と欠陥のつながりが発生した回数

(3)欠陥の重みは、対象欠陥に連結する過失因子-欠陥エッジの重みの合計とする。

② 手順①で作成した重み付き有向グラフに対して、検出した欠陥よりも共起の観点で潜在確率の高い欠陥を抽出できることを確認する。抽出条件は以下の通りとする。

【抽出条件】

各欠陥ノードにて、以下の条件を満たすノードを抽出する。

条件式) 検出した欠陥ノードの重み \leq 欠陥ノードの重み

4.3.1 実験2の例

実験手順①：実験1の手順③で作成したグラフに重み付けをする。

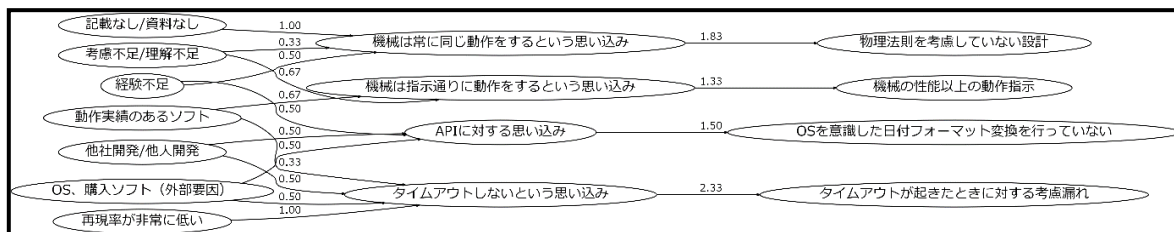


図5. 重みを付加した有向グラフ

上記の例では、欠陥予測DBに過失因子「機械は指示通りに動作するという思い込み」が2件登録されている。このため、上記過失因子と、誘発因子「考慮不足/理解不足」、「動作実績のあるソフト」は4.3章の重みの算出方法(1)に従い、「0.67」となる。

実験手順②：上記のグラフの欠陥「OSを意識した日付フォーマット変換を行っていない」を検出した欠陥と仮定し、自身の重み(1.50)以上の欠陥を抽出する。

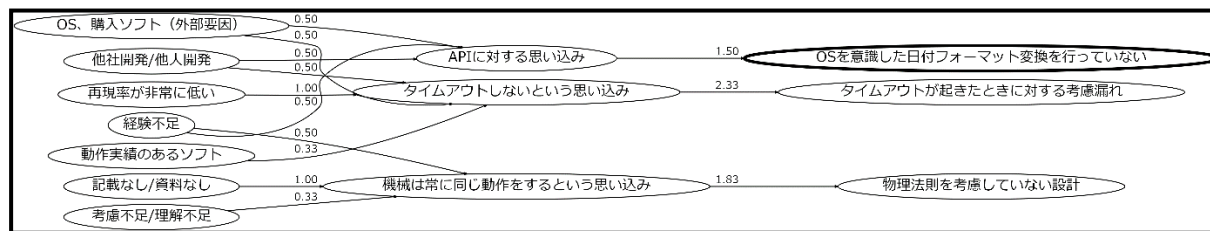


図 6. 抽出アルゴリズム適用後に抽出された欠陥

上記の結果、図 5 に存在する欠陥「機械の性能以上の動作指示」(重み 1.33)は、検出した欠陥「OS を意識した日付フォーマット変換を行っていない」よりも潜在する可能性が低いと判断し、候補から外す。

4.3.2 実験 2 結果

実験 2 として、検出欠陥よりも潜在する可能性が高い欠陥を推定し、推定数の変化を確認した(付録 8 参照)。その結果、因子間の重み(関連強さ)を用いることで、定性的な推定に比べて、共起性の高さを考慮した上で推定数の絞り込みを行えることが確認できた。これにより実験 1 の問題点が解消され、COP アプローチの有用性が向上したと考える。

なお一方で、推定数に変化が見られない欠陥の存在も確認した。これは、欠陥の中で因子間の重みが小さい欠陥によく見られた。本実験では重みを用いて、より潜在する可能性が高いものを推定対象としたため、重みが小さいと推定数が多くなる傾向となると推察する。これらの欠陥については、重みが小さい(=発生頻度が低い)ため、推定された全ての欠陥の確認を行うのではなく、個別に対応を行うなどの判断が必要であると考えられる。

4.4 考察

本研究の実験結果に対して、考察を次に示す。

RQ1：欠陥にはどの程度共起性があるのか？

欠陥モデルをもとに、欠陥混入のメカニズムについてグラフ理論を用いて検証した結果、多くの欠陥に共起性があることを確認できた。この結果、欠陥検出時の欠陥情報をもとにした定性的な潜在欠陥の推定が可能であることを示せた。

RQ2：欠陥の共起性を利用した、普遍的な潜在欠陥の見つけ方はあるのか？

本研究では、研究員が所属する企業の欠陥情報をもとに、グラフ理論を適用して検証を行った。その結果、共起性を利用した潜在欠陥の推定を行うことで、二つの企業で推定ができることを確認できた。これにより、COP アプローチは、グラフというよく知られた数学モデルを用いることで、業務ドメインに依存しない普遍的な共起欠陥の見つけ方のひとつとなると考えられる。

また、共起性だけでなく因子の出現回数をもとにして潜在確率の高い欠陥を推定することで、定量的な潜在欠陥の推定を可能とした。今回は一例として潜在している可能性が高い欠陥に絞り込むことを行ったが、逆に可能性が低い欠陥に絞り込むことも可能である。例えば、潜在している可能性が低い欠陥であっても、重大な事象を引き起こす欠陥に対しては対策を検討するといったケースが考えられる。

実験では、分析の起点となる欠陥とは一見関連性の無い欠陥も推定された。この結果は、人間の主観的な判断では推定が難しい欠陥も推定できる可能性を示唆していると考えられる。ただし、実験 1 の結果で見られたように、「～不足」というどのような欠陥にも当てはまり得る誘発因子が含まれると、多数の欠陥に共起性が見られることになる。この結果から、推定の精度には、入力となる欠陥分析の結果が大きく影響することが確認された。このため推定の精度を高めるには、全体的な欠陥分析の質の向上が不可欠であると考えられる。

第7分科会 (Team KuKuRu)

全体を通して、グラフを用いた共起欠陥の推定をすることで、複雑な欠陥発生が絡み合う品質の全体像把握が可能になると考える。また、作成したグラフは、実際に人間の思考の中にある欠陥の経験や知識の繋がりであると捉えた。このように考えた場合、実際の開発者や品質技術者は、複雑な直感に基づいて事象を辿り、広範囲な欠陥分析を行っていたであろうことが想像できた。今回の提案により、人間の思考の中というブラックボックスからの直感による欠陥分析ではなく、可視化された情報を基にした欠陥分析を行える可能性を示せたと筆者らは考える。

4.5 妥当性への脅威

本実験では、研究員の主観により実験データを選択しているため、何らかのバイアスがかかっている可能性がある。また、サンプル数が少ないことが結果に影響を与えている可能性もある。

5 おわりに

5.1 まとめ

本研究では、開発工程で検出した欠陥と過去の欠陥を、欠陥混入のメカニズムの関連性を用いて照合し潜在欠陥を推定する「COPアプローチ」を確立し、有効性を確認した。

また追加施策によって、誘発因子や過失因子と欠陥の関連性の強さを考慮することで、より潜在している可能性の高い欠陥を推定できることを確認した。これにより、欠陥を効率的に一網打尽で除去し、後工程への欠陥流出抑制につながる手法の提案が行えた。

5.2 今後の展望

今回は過去プロジェクトの欠陥情報の一部を用いての実験であった。そのため、実際に開発中のプロジェクトにCOPアプローチを適用して、後工程への欠陥流出抑制効果を確認することを今後の課題としたい。

今回実験で行ったように、推定用欠陥モデルはプログラムにより生成できるため、データとして欠陥情報を蓄積して扱うことができる。さらに、出現回数をもとにした重みを付け加えることで、リアルタイムかつ定量的に対象プロジェクト(企業も含む)の状況を認識することが可能と考える。地道に欠陥情報を蓄積していくことで、本研究が「欠陥の推論エンジン」の実現に繋がることを期待する。

参考文献

- [1] 細川宣啓, 西康晴, 嬉野綾, 野中誠, 原佑貴子, 「過失に着目した欠陥のモデリングーバグ分析はなぜうまくいかないのか?ー」, ソフトウェアテストシンポジウム 2013 東京, 2013
- [2] 細川宣啓, 永田敦, 柏原一雄, 岡本晃, 鈴木裕一郎, 田村光義, 東久保理江子, 保栖真輝, 「ソフトウェア欠陥予測アルゴリズム」, 日本科学技術連盟 SQiP 研究会, 2014
- [3] 小島 義也, 「評価者による ODC を使用した不具合分析の現場展開～属人化を排除していく試み」, ソフトウェアテストシンポジウム 2015 東京, 2015
- [4] 安藝優子, 野中誠, 「摘出済み欠陥数を考慮したレイリーモデルに基づく ソフトウェア欠陥予測手法」, 経営情報学会 全国研究発表大会要旨集 2010, 2010
- [5] Jean-Loup Guillaume, Matthieu Latapy, Bipartite graphs as models of complex networks, Physica A: Statistical Mechanics and its Applications, Volume 371, Issue 2, Pages 795-813, ISSN 0378-4371, 15 November 2006, 2006
- [6] AT&T 研究所, Graphviz - Graph Visualization Software, <http://www.graphviz.org/>, アクセス日:2017/1/16