

# 不確かさの発生過程に関する実証分析

村岡 北斗<sup>1,a)</sup> 深町 拓也<sup>1,b)</sup> 山下 一寛<sup>1,c)</sup> 鵜林 尚靖<sup>1,d)</sup> 亀井 靖高<sup>1,e)</sup>

概要：ソフトウェア開発において「不確かさ」は避けることができない問題である。我々はこの問題の解決に向けて、その発生から修正までの流れについて着目し、その特徴を得るために改良 SZZ アルゴリズムを用いて不確かさの発生を特定する方法を提案する。不確かさが発生したコミットからその特徴を獲得し、不確かさを抱擁した開発の助けにすることを目標とした。本稿では、予備実験の結果と今後の研究の方向性を示す。

## 1. はじめに

ソフトウェア開発の様々な段階において「不確かさ」というものは避けられないものである [3]。従来のソフトウェア工学では、「不確かさ」に対するサポートはリスク管理などの方法論に留まっており、設計や実装・テスト技術の中では明示的に扱われて来なかった。今後は、「不確かさ」を抱擁（あるいは包容）するモデル駆動開発支援系、プログラミング言語処理系、検証系、欠陥予測モデルなどを提供していくことがソフトウェア工学の研究上重要になっていくものと思われる。

本研究では、この不確かさの特徴や原因などを分析するために、不確かさがどのように発生しどのように対処されているのかを調査した。不確かさのキーワードを含むログメッセージに含んでいるものをリストアップし、SZZ アルゴリズム [4,5] の一部を使用することで不確かさの発生を突き止めその傾向を掴む手がかりとした。

## 2. 本研究のアプローチ

不確かさの問題を解決するために不確かさの発生に着目した。今まで不確かさの発生原因については明らかではなかった。本研究では不確かさの発生を明らかにするため SZZ アルゴリズムを改良し使用した。SZZ アルゴリズムとは、バグの発生を追跡するアルゴリズムである。このアルゴリズムは大きく 2 つのステップから成り立つ。1 つ目のステップとして、バージョン管理システムの履歴からバグ

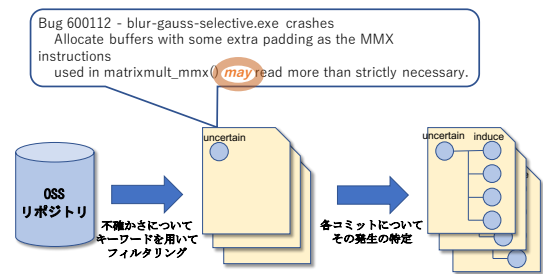


図 1 本研究の流れ

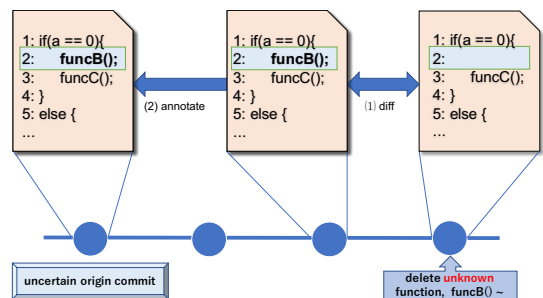


図 2 不確かさの発生特定手法

に関するキーワードがコメントに存在するかなどの指標を用いてバグが修正されたコミットを見つけ出す。2 つ目のステップでは、バグの修正前後のコミットについて各ファイルと比較する。そこで変更された行をバグ修正箇所とし、その行がどのコミットで追加されたかを特定する。

本研究では、バグのコミットを探すステップで対象をバグの代わりに不確かさへ置き換えることで、不確かさの発生コミットを特定することにした (図 1)。不確かさに関するキーワードをコメントに含んでいるかどうかを、コミットが不確かさを持つものであるかの判断基準とした。今回特定に用いたキーワードは図 3 のものを使用した。このキーワードは Oxford American Writer's Thesaurus から参照した uncertainty に関する類義語である。不確かさを

<sup>1</sup> 九州大学

Kyushu University

a) muraoka@posl.ait.kyushu-u.ac.jp

b) fukamachi@posl.ait.kyushu-u.ac.jp

c) yamashita@posl.ait.kyushu-u.ac.jp

d) ubayashi@ait.kyushu-u.ac.jp

e) kamei@ait.kyushu-u.ac.jp

表 1 ログメッセージの例

No.	キーワード	ログメッセージ
1.	Unknown	The file builds now and I only see warnings about using <u>unknown</u> Carbon API.
2.	Unknown	Also remove some junk that was there for <u>unknown</u> reasons, this tool has a long history.
3.	Unclear	It is highly <u>unclear</u> when to return FALSE.
4.	Debatable	Whether or not undo memory should be included here is <u>debatable</u> .
5.	Unsure	This needs to be done in the makefile.msc too, but I'm <u>unsure</u> on the right way to do that.

debatable, undetermined, unsure, unpredictable, unforeseeable, incalculable, risky, chancy, dicey, informal, iffy, vague, ambiguous, unknown, unascertainable, obscure, arcane, changeable, irregular, unreliable, unsettled, erratic, fluctuating, doubtful, dubious, undecided, irresolute, vacillating, unclear, ambivalent, hesitant, tentative, faltering, unconfident, may, might, probably, fuzzy

図 3 uncertain に関するキーワード一覧

表 2 GIMP [1] のプロジェクト概要

全コミット数	期間	コミッタ数
37,713 件	1997/6/1-2016/10/25	375 人

含むとして抽出したコミット群に SZZ アルゴリズムの 2 つ目のステップを適用することで不確かさの発生コミットを特定した (図 2)。このコミットの内容を解析することで不確かさ発生に関する特徴が存在するか調査した。

### 3. 予備実験

予備実験で使用したデータセットを表 2 に示す。その結果として出てきた不確かさが発見されたコミットから特定した不確かさの発生コミットの中に、不確かさが発生する原因となる特徴を探し出そうとした。全コミット 37,713 件のうちキーワードを用いて特定した不確かさを含むコミットは 479 件存在した。一部を表 1 に示す。また、ここから特定した不確かさの発生コミットと対応させた発生源のペアが 13,557 件存在した。つまり、1 つの不確かさを含むコミットに対して平均 28.3 件の発生源となったコミットが存在した。予備実験で特定した不確かさを持つあるコミット\*1のログメッセージの中に、長年の更新によって分からなくなってしまった記述を削除したという更新があった (図 4)。その削除された部分の記述は書かれてから最長で 9 年経過しており、その削除した記述は削除した本人が追加したものが半分以上を占めていた。そのためこの不確かさについては、自分が書いたはずのコードを長年経過したため忘れてしまったため生まれた不確かさであると予想できる。この例からは、同一のファイルを複数回更新することで不確かさが発生しやすくなるのではないのかという仮説を立てることができる。

\*1 <https://github.com/GNOME/gimp/commit/f078a7416c163e743bd19f6f5c0a250a08e8c4c8>

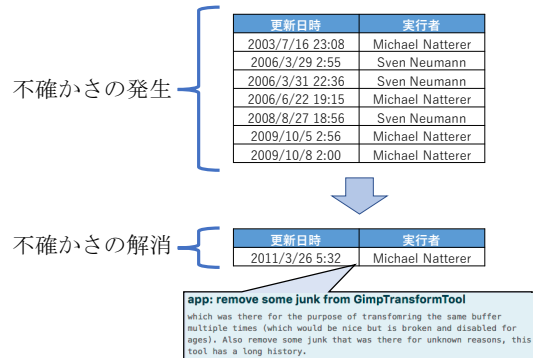


図 4 不確かさを含むコミットの分析例

### 4. 今後の予定

現在対象としている OSS の数が少ないため、多くの OSS について SZZ アルゴリズムを適用し、今回立てた仮説の検証を行う。また、不確かさに関するキーワードごとに特徴が見つからないかについても調査したい。最終的に我々が開発しているインターフェースを用いて不確かさのマネジメントを行うツール iArch-U [2] に機能として取り込むことを考えている [6]。

### 謝辞

本研究は、文部科学省科学研究補助費基盤研究 (A) (課題番号 26240007) による助成を受けた。

### 参考文献

- [1] GIMP, <https://www.gimp.org/>.
- [2] iArch, <https://posl.github.io/iArch/>.
- [3] Garlan, D.: Software engineering in an uncertain world, *Proceedings of the FSE/SDP workshop on Future of software engineering research*, ACM, pp. 125–128 (2010).
- [4] Kamei, Y., Shihab, E., Adams, B., Hassan, A. E., Mockus, A., Sinha, A. and Ubayashi, N.: A large-scale empirical study of just-in-time quality assurance, *IEEE Transactions on Software Engineering*, Vol. 39, No. 6, pp. 757–773 (2013).
- [5] Śliwerski, J., Zimmermann, T. and Zeller, A.: When do changes induce fixes?, *ACM sigsoft software engineering notes*, Vol. 30, No. 4, ACM, pp. 1–5 (2005).
- [6] 深町拓也, 鵜林尚靖, 細合晋太郎, 亀井靖高ほか: Git 連携による不確かさマネジメントシステム, ソフトウェアエンジニアリングシンポジウム 2016 論文集, Vol. 2016, pp. 70–77 (2016).