

令和元年6月4日現在

機関番号：12608  
研究種目：基盤研究(C) (一般)  
研究期間：2014～2018  
課題番号：26330077  
研究課題名(和文) 言語独立・ハッシュ値・行粒度の追跡子による，ソフトウェア追跡性の高精度な確保

研究課題名(英文) Maintaining high-accurate software traceability using language-independent/hash-based/line-based tracers.

研究代表者  
権藤 克彦 (Gondow, Katsuhiko)  
東京工業大学・情報理工学院・教授

研究者番号：50262283  
交付決定額(研究期間全体)：(直接経費) 3,800,000円

研究成果の概要(和文)：ソフトウェア開発では軽量かつ効果的な追跡性リンクの確保が重要である。しかし既存研究の多くは事後に情報検索技術を用いて精度が悪いという問題があった。事後アプローチとは異なり、本研究では新しい軽量な事前アプローチ(TCCと呼ぶ)を提案した。TCCは追跡子(グローバルID)として、Gitなどの分散版管理システムが用いるハッシュ値を使う。TCCはテキスト文字列のコピー&ペースト時に、コピー内容に自動的に追跡子を埋め込むため、利用者が明示的に追跡子を埋め込む必要はない。また、その際に参照先のデータをローカルリポジトリに格納するので、URLのリンク切れの際も保守時に参照先の内容を閲覧できる。

#### 研究成果の学術的意義や社会的意義

現代社会は多くの場面でソフトウェア技術に支えられているが、しばしばソフトウェア障害による数十億円規模の莫大な損害が出たり、場合によっては人命が危機にさらされる。本研究が提案した追跡性の技術は「ソフトウェアのある部分なぜこうなっているのか」を辿りやすくし、ソフトウェアが正しいことを確認したり、誤りの発見を支援する技術である。莫大な損害や開発費の大幅な削減に貢献する。

研究成果の概要(英文)：In software development, it is crucially important to effectively capture, record and maintain traceability links in a lightweight way. Most of previous work are retrospective approaches based on information retrieval techniques, but they are likely to generate many false positive traceability links. Unlike retrospective approaches, this work proposes a novel lightweight prospective approach, which we call TCC (tracer-carrying code). TCC uses a hash value as a tracer (global ID), widely used in distributed version control systems like Git. TCC automatically embeds a TCC tracer into source code as a side-effect of users' copy&paste operation, so users have no need to explicitly handle tracers (e.g., users have no need to copy&paste URLs). TCC also caches the referred original text into Git repository. Thus, users can always view the original text later by simply clicking the tracer, even after its URL or file path is changed, or the original text is modified or removed.

研究分野：ソフトウェア工学

キーワード：ソフトウェア保守 ソフトウェア追跡性 追跡子 ハッシュ値

## 様式 C-19、F-19-1、Z-19、CK-19 (共通)

### 1. 研究開始当初の背景

莫大なソフトウェア開発コストの大部分がソフトウェア保守費用であり、ソフトウェア保守作業の多くがソフトウェア理解であると言われている。特に、例えば、あるソフトウェアモジュールのソースコードが、どの仕様書を参照して記述されたかなどを知るためのソフトウェア追跡性は非常に重要である。ソフトウェア追跡性を効率よく効果的に確保・記録・保守することができれば、ソフトウェア保守費用を大幅に削減できるようになる。既存研究ではソフトウェア追跡性を十分には確保できていない。

これまでの主な既存研究は、人手による手動で追跡性を確保する、あるいは単語の類似性を用いて事後に自動的に追跡性リンクを回復する研究が多く、手間がかかる、精度が悪いという問題があった。例えば、追跡の対象となる仕様書や法律が何度も改訂された場合、単語の類似性に基づく技術では、これらの改訂された複数の版をすべて検出してしまう。また単語として出現しない場合は未検出となってしまう。例えば「UNIX シグナルハンドラ中で `fprintf` を使っていない」という情報を追跡子として埋め込みたい場合を考えると、ドキュメントには「UNIX シグナル中で使っている良い関数のリスト表」が記述されているので、単語の類似性では未検出となる。

### 2. 研究の目的

ソフトウェアの追跡性とは「ソフトウェアが満たすべき要求仕様などが、実装コードのどこに反映されたかを確認できる性質」である。莫大なソフトウェア開発コストの大部分がソフトウェア保守費用であり、ソフトウェアの追跡性の確保により、ソフトウェア保守費用を大幅に削減できるため、ソフトウェアの追跡性は非常に重要である。しかし、既存研究ではプログラムの高い可読性を保持しつつ、ソフトウェアの高い追跡性の確保することは難しい。本研究ではプログラムのハッシュ値を追跡子とする手法を新たに提案することを目的とする。単語の類似性を用いた解析的手法では精度が悪いが、ハッシュ値では1文字でも異なればほぼ確実に異なるハッシュ値となり高精度化が可能である。また、コピー&ペーストに付随したリンク挿入が可能のため、ハッシュ値の追跡子は挿入・保守のコストも安い。本研究の目的は以上の追跡子を提案・実装し、ソフトウェア追跡性の高精度な確保することを目的とする。

### 3. 研究の方法

まず提案する追跡子の要件として、以下を挙げた：事前に手動で軽量にリンクできること、位置独立で言語独立であること、永続性を持つこと、情報の一体性を持つこと、文脈独立であること、処理を用意にするため行単位でテキストベースであること（コピー&ペーストが容易にできること）、その他として保守性や可読性をなるべく損なわないこと。

ハッシュを識別子として使用するリポジトリとして分散型版管理ツール **Git** を用い、ユーザのコピー&ペースト操作に連動させて、(1) 元のドキュメントを自動的に **Git** リポジトリに格納し、(2) 元ドキュメントのハッシュ値をソース・コード中に埋め込む、というツール **TCC** を上記要件をなるべく満たすように提案実装する。実装したツールを用いて簡単なソフトウェア開発を行い、**TCC** の有用性を評価する。

### 4. 研究成果

3で述べた軽量の追跡子ツール **TCC** のプロトタイプ実装を行った。ハッシュ値には20バイト長のSHA-1を、リポジトリとして分散型版管理システム **Git** を用いた。プロトタイプ実装を行った環境は次の3つである：エディタ実装 (Emacs版)、Webブラウザ実装 (Chrome 拡張、参照のみ)、PDFビューワ実装 (Chrome 拡張、参照のみ)。これらの実装には、単なるハッシュ値の埋め込みのみではなく、ペアハッシュと部分ハッシュの実装も行った。ペアハッシュは仕様書とソースコード両方のハッシュをペアで管理することで、意図しない片方のみの更新を自動的に検出する。部分ハッシュはハッシュ計算の対象をドキュメントの一部に絞ることで、対象外の変更を無視する機能である。

この **TCC** のプロトタイプ実装を用いて、小さな iPhone アプリの開発を我々自身で行った。実装根拠の記録、実装中の疑問や問題の記録に有用であるという結果

を得た。例えば「実装上のノウハウとして、歩数を記録するためにはGPS サービスAPIも使用する必要がある」という情報をTCCの追跡子として保存することができた。

また、この研究を通じて、追跡子に関する周辺技術について例えば以下の研究成果を出すことができた。

・電子画像データに埋め込む追跡子 RIC (robust index code) : 電子画像データ中に埋め込む追跡子としてRIC (robust index code)という新たなカラーコードを提案実装した。通常のSNSへのアップロード時の発生する圧縮に対して、RICでは情報が喪失しない(頑健である)ことを示した。また、RICの超流通システム(super distribution system)としての有用性を議論し、RICを利用した画像超流通システムは、ラップが削除可能であるが、追跡子に紐付けたコンテンツは複製不可能であること、AmazonやNetflixなどのコンテンツ・プラットフォーム方式と比較してアクセス容易性や著作権者の利益の点で優れることなどを明らかにした。

・プログラム開発中のコード断片を、編集中のプログラムの表情や独り言に紐付ける追跡技術 CodeCAM : 既存のツールやIDEへの変更が不要で、かつ言語独立に実装するために、キーマシンを用い、その入力列とコード断片中を紐付ける新しい方法を提案実装した。予備実験では、殆どの時間は無表情だが、起こした表情の中では「幸せ」と「驚き」が主であり、表情とアクションに相関があるという結果を得た。

## 5. 主な発表論文等

[雑誌論文] (計1件)

- ① M. Kim, K. Lee, K. Gondow, J. Imura: Robust Index Code to distribute digital images and digital contents together, IEICE TRANSACTIONS on Information and Systems, E101.D(9):2179-2189, DOI: 10.1587/transinf.2017PCP0004 (2018) 査読あり

[学会発表] (計6件)

- ① M. Kim, K. Lee, K. Gondow, J. Imura: Robust Index Code with Digital Images on the Internet, 13th Int. Conf. Signal Processing and Multimedia Applications (SIGMAP2016), pp.28--37 (2016) 査読あり
- ② 権藤克彦, 荒堀喜貴, 山本晃治, 福寄雅洋, 梅川竜一: ハッシュ値を用いた軽量追跡ツール, 電気情報通信学会 技術研究報告 信学技報, vol.116, no.425, pp.13--18 (2017) 査読なし
- ③ M. Kim, K. Lee, K. Gondow: Distribute digital contents within digital images on the mobile, 9th Int. Conf. on E-Education, E-Business, E-Management and E-Learning (IC4E 2018), pp.35--39 (2018) 査読あり
- ④ Y. Shinyama, Y. Arahori and K. Gondow: CodeCAM: capturing programmer's reaction during coding session, 2018 IEEE Workshop on Mining and Analyzing Interaction Histories (MAINT), pp. 1-5 (2018). doi: 10.1109/MAINT.2018.8323087 査読あり
- ⑤ K. Gondow, Y. Arahori, K. Yamamoto, M. Fukuyori and R. Umekawa: TCC (Tracer-Carrying Code): A Hash-based Pinpointable Traceability Tool using Copy&Paste, 13th Int. Conf. Software Technologies (ICSOFT 2018), pp. 221-232 (2018) DOI:10.5220/0006837102210232 査読あり
- ⑥ Y. Shinyama, Y. Arahori and K. Gondow: Analyzing Code Comments to Boost Program Comprehension, 25th Asia-Pacific Software Engineering Conf (APSEC 2018), 10 pages, (2018) 査読あり

[図書] (計0件)

〔産業財産権〕

○出願状況（計 0 件）

名称：  
発明者：  
権利者：  
種類：  
番号：  
出願年：  
国内外の別：

○取得状況（計 0 件）

名称：  
発明者：  
権利者：  
種類：  
番号：  
取得年：  
国内外の別：

〔その他〕

ホームページ等

TCC プロトタイプ実装 <http://www.sde.cs.titech.ac.jp/tcc/>

## 6. 研究組織

### (1) 研究分担者

研究分担者氏名：  
ローマ字氏名：  
所属研究機関名：  
部局名：  
職名：  
研究者番号（8 桁）：

### (2) 研究協力者

研究協力者氏名：  
ローマ字氏名：

※科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。