開発者が行ったソースコード修正作業を学習し代行するボット
― DevReplay：開発者を再現する Lint ―

1.　背景

ソフトウェア開発において，デバッギング作業はソフトウェアの信頼性確保に重要であると同時にコストの高い作業でもある．修正作業も含めたデバッギングは開発作業全体で約50%の時間的コストを開発者が消費すると最近の研究で報告されている．

デバッギング作業にコストがかかる最も大きな原因として，問題の特定と修正を開発者が広範囲かつ手作業で行う必要がある点が挙げられる．本プロジェクトでは修正コストの削減を目的に申請者は開発者が行ったソースコード修正作業を記録し，過去に行った作業を開発者の代わりに行うツール「DevReplay」を開発，公開した．

2.　目的

本プロジェクトは，過去に開発者が行ったソースコードの修正を自動的に行うことで時間的コストを削減することを目的とした．

そのために GitHub で公開されているソフトウェアプロジェクトから，ソースコードの開発履歴を収集し，他人が行った修正方法を自分のソースコードに適用する開発環境を構築することとした．これにより，開発者にとって道の問題がソースコードにあったとしても，関連プロジェクトから修正方法を知ることができる．

例えば，実行速度に優れたソフトウェアの開発履歴を利用して，自身のソースコードをパフォーマンスに優れたものに改善することができる．また，収集した開発履歴を転用することで，新規ソフトウェアやプログラミング教育のように開発履歴を持たないソフトウェア開発の場合でも，ソースコードの改善を行えるようになる．

3.　開発の内容

本プロジェクトではソースコード修正の提案や自動修正を行う開発環境を実現した．具体的には，GitHub 上で公開されているソフトウェアの開発履歴からソースコードの改善パターンを発見し，不特定多数に共有可能なファイルとして出力する「パターン発見ツール」を開発した．また，改善パターンに基づいたソースコードの改善を自動的に行うソースコード修正ツールを開発した．

図 1 に Git を用いた場合のパターン生成プロセスを示す．まず対象となるソフトウェアリポジトリから変更前変更後のコミットを得る．次に変更前，変更後のソースコードを抽象構文木に展開する．このとき変更前と変更後で共通している識別子や数字は$1 や${2:num}という形に抽象化する．最後に抽象化し，識別子も含めた TextMate Snippet を JSON ファイルとして出力する．

DevReplay は Visual Studio Code など Language Server Protocol を用いたエディタプラグイン，GitHub Application，コマンドラインツールとして利用することができる．ユーザが DevReplay を利用したいプラットフォームにこれらのアプリケーションをインストールし，プロジェクトにパターンファイルを「devreplay.json」として保存することで，修正すべき箇所に警告が出力されるようになる．

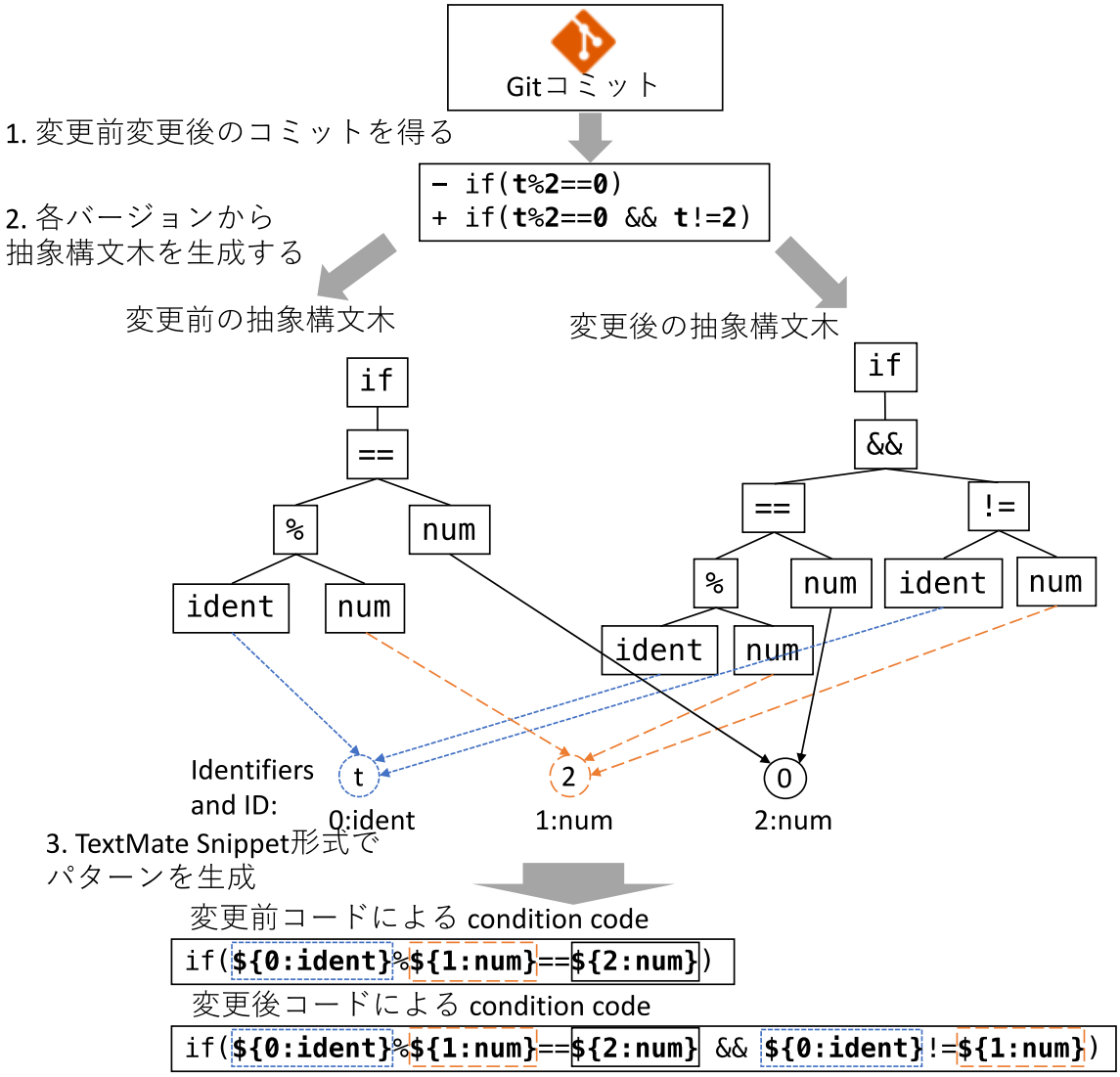

図 1. Git を用いた場合のパターン生成プロセス

実際に Visual Studio Code 上での Python のソースコードを，パターンに従って改善した例を図 2，図 3 に示す．

パターンの基本要素は上述のパターン発見ツールで自動生成される変更前後のスニペット“condition”，“consequent”で構成される．この例では右側のパターン condition に一致するソースコードを波線で警告する．追加要素として，説明を示す“description”，重要度を示す“severity”，作者や情報源を示す“author”を利用者が記述することで，エディタ上の表示を利用者の感覚に合わせることができる．図 3 はパターンに従って 2 つの修正を行った例である．一つ目の修正は 2 行目で if 文の曖昧さを取り除いている．もう一つは 8–10 行目の変数のスワップを同等の 1 行の記述に改善している．修正は図 2 の左側の 1 行目に表示されているような電球のアイコンをクリックすることで自動的に行われる．

現在 DevReplay は C, C++, Java, Dart, JavaScript, Python, Go, TypeScript, COBOL, Ruby, PHP, R の 12 言語に対応しており，今後も利用者の要望に応じて追加する予定である．

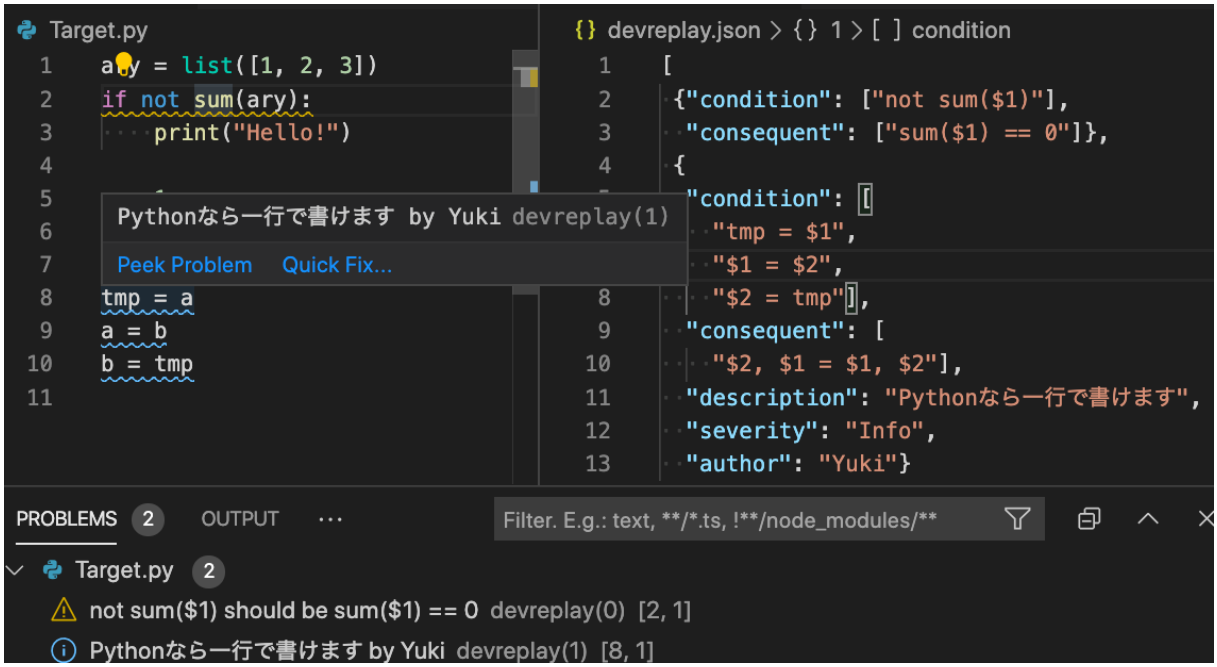

図 2. Visual Studio Code における利用例

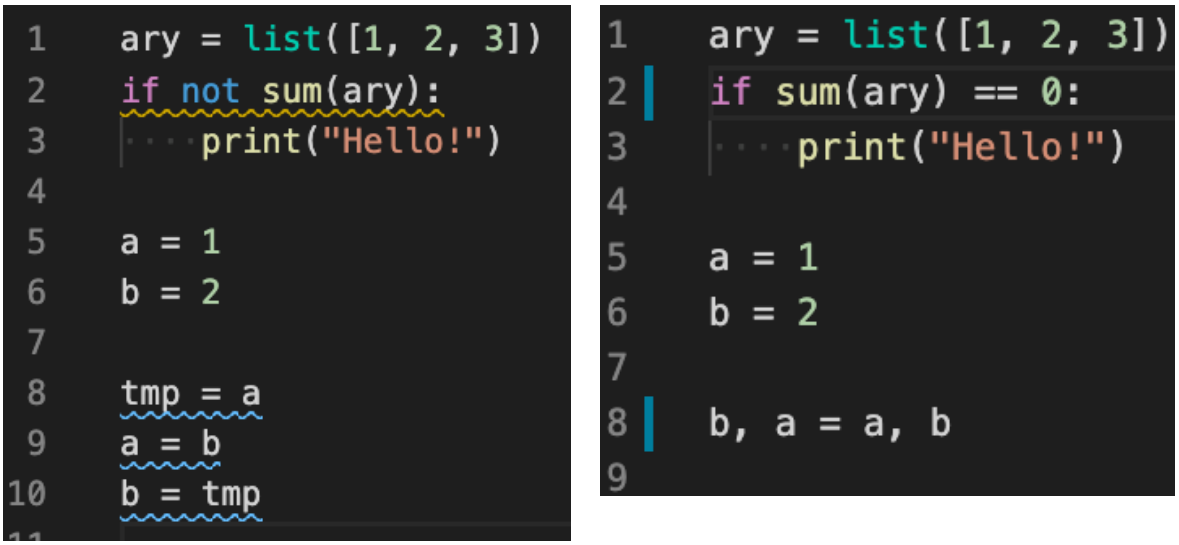

図 3. DevReplay による修正前（左）と修正後（右）の例

4. 従来の技術（または機能）との相違

従来技術として，コーディング規約に違反したソースコードの指摘を行う静的解析ツールや，あらかじめ作成されたテストコードにソースコードの動作がパスするように修正する自動プログラム修正ツールが存在する．

本プロジェクトの特徴は，プロジェクトの暗黙的なソースコード改善パターンを新たに発見する点である．ソフトウェアを構成するソースコードの記述内容は，そのときの採用技術や開発チームの方針によって変化する．本プロジェクトはソースコードの修正履歴から，固有名詞や社内ルールのような開発チームの中で閉じた修正も自動的に実施可能にする．

5.　期待される効果

ソフトウェア開発履歴に基づきソースコードを自動修正する本システムを用いることによって，ソフトウェア開発者が容易に巨人の肩の上に立てる開発プラットフォームを実現できる．

本プロジェクトの応用として，ソースコードを評価することで，開発者の教育に利用することができる．経済産業省の報告によると IT 人材は国内で 17 万人不足しており，開発人材の確保が重要である．一方で，「良いコード」，「良い開発者」を定量的に評価する手法は少ない．本プロジェクトは，上級者（有名な開発者や講師など）の書いたソースコードを正解集合として，類似したソースコードを記述する人材を評価できる．

また，セキュリティ脆弱性の対応を早めることができる．現在セキュリティ上に影響の大きな脆弱性が発覚した場合，開発者はソフトウェアごとに対応を行う必要がある．本プロジェクトを利用することで，自分のソフトウェアと類似した機能を持つソフトウェアがセキュリティに関する対応を行った際に自らのソフトウェアのソースコードにも同様の対応するよう通知することができる．

6.　普及（または活用）の見通し

今後は国内外の学会にて発表やデモセッションに参加する．また，ツールの認知度を向上させるために，TensorFlow や PyTouch など多くの開発者が利用する技術に特化したパターンを公開していく．

7.　クリエータ名（所属）

上田　裕己（奈良先端科学技術大学院大学　ソフトウェア工学研究室）

（参考）関連 URL

- DevReplay 配布サイト：https://devreplay.github.io/
- DevReplay プロジェクトのソースコード：https://github.com/devreplay
- DevReplay 本体：https://github.com/devreplay/devreplay
  - コード改善パターン生成スクリプト群：
    https://github.com/devreplay/devreplay-pattern-generator
  - ソースコードのトークン化および差分作成スクリプト：
    https://github.com/Ikuyadeu/CodeTokenizer
  - DevReplay Visual Studio Code プラグインおよび Language Server：
    https://github.com/devreplay/vscode-devreplay
  - DevReplay Atom エディタプラグイン：
    https://github.com/devreplay/atom-devreplay
  - DevReplay GitHub アプリケーション：
    https://github.com/devreplay/github-app-devreplay
  - DevReplay Web ページ：https://github.com/devreplay/devreplay.github.io