

言語サーバを応用した細粒度編集履歴 収集プラットフォームの構想

石田 直人¹ 神田 哲也^{1,a)} 嶋利 一真^{1,b)} 井上 克郎^{1,c)}

概要：ファイルの保存、版管理システムへのコミットといった単位よりも細かい、エディタの操作履歴のような細粒度編集履歴は、プログラムの理解支援や変更支援、プロセス改善などに活用されている。本稿では、様々なエディタが用いられるような状況でも細粒度編集履歴を収集できるようにするために、細粒度編集履歴の収集部分を言語サーバを応用したモジュールに分離し、多様な開発環境での情報収集と活用を可能にするためのプラットフォームを提案する。

1. はじめに

ファイルの保存、版管理システムへのコミットといった単位よりも細かい、エディタの操作履歴のような細粒度編集履歴を記録し、活用する研究がなされている。細粒度編集履歴は、プログラムの理解支援や変更支援、プロセス改善などに活用されている [1]。教育機関でのプログラミング演習においても、学生の進捗や理解度、行き詰まりを把握するためにこのような履歴を収集分析することが提案されている。

細粒度編集履歴の収集のためには、開発環境においてそれを取得する機構を組み込む必要がある。これは、もし複数の開発環境に対して細粒度履歴収集を行いたい場合、それぞれにおいてログ取得の機構を準備する必要があるということでもある。StackOverflow Developer Survey 2019 [2] の「Most Popular Development Environments」によると、使用率が 10% を超えるエディタが 10 あることから、開発者が使用するエディタは多岐にわたり、また一人の開発者が複数のエディタを使い分けることも珍しくないことが読み取れる。教育現場でも、卒業までに複数の言語にふれるようなカリキュラムの場合、複数種類の開発環境を使うことが想定される。

そこで本稿では、多様な開発環境での情報収集と活用を可能にするために細粒度編集履歴の収集部と分析部を分離したプラットフォームを提案する。具体的には、細粒度編集履歴の収集部分を多くのエディタで利用可能な言語

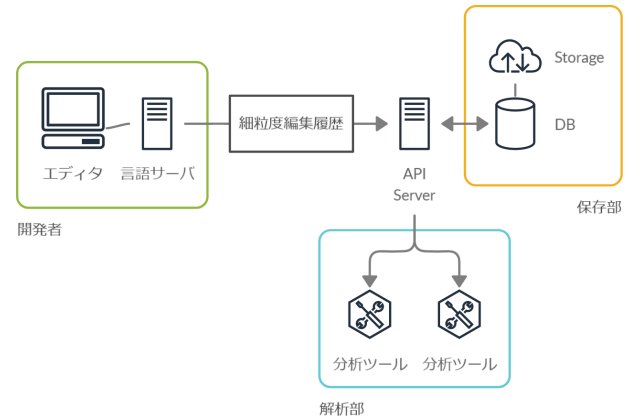


図 1 細粒度編集履歴収集プラットフォームの構成

サーバを応用したモジュールに分離することで、種類の異なるエディタでも容易に情報収集が行えるようにする。また、分析部はサーバ上に構築することで、開発環境によらない細粒度編集履歴の利活用を可能にする。

2. 細粒度編集履歴収集プラットフォーム

今回、細粒度編集履歴収集のプラットフォームを提案するにあたり、以下の 2 つの要件を定めた。

- (1) 開発者が好みのエディタを使用できる (特定のエディタに依存しない)
- (2) 細粒度編集履歴の分析ツールや、CI ツールなどの開発者向けツールと容易に連携することができる

これらの要件を満たすために、エディタ側では Language Server Protocol [3] (以後 LSP とする) を利用し細粒度編集履歴の収集を行う。また、収集した情報を保存、分析するための基盤はサーバとして実装し、サーバがデータ取得用の API を提供することで他のツールとの連携を容易に

¹ 大阪大学
Osaka University, Suita, Osaka 565-0871, Japan
a) t-kanda@ist.osaka-u.ac.jp
b) k-simari@ist.osaka-u.ac.jp
c) inoue@ist.osaka-u.ac.jp

```

{
  "type": "edit",
  "time": "2020-08-04T08:11:10.608Z",
  "payload": {
    "textDocument": {
      "uri": "file:///e%3A/gprac/src/
        components/header.js",
      "version": 15
    },
    "contentChanges": [
      {
        "range": {
          "start": {
            "line": 3,
            "character": 11
          },
          "end": {
            "line": 3,
            "character": 15
          }
        },
        "rangeLength": 4,
        "text": "ABCD"
      }
    ]
  }
}

```

図 2 収集部から送信される細粒度編集履歴の例

することを考える。図 1 に提案手法の構成図を示す。収集部である開発環境と分析部であるデータの保存、解析はそれぞれ独立しており、API サーバによって連携される。

2.1 細粒度編集履歴の収集部

種々の開発環境における細粒度編集履歴の収集を実現するため、LSP を応用する。LSP とは、エディタと言語サーバ間のやり取りを行うためのプロトコルであり、「コード補完」や「定義へ移動」といった機能を言語サーバがサービスとして提供し、LSP を通じエディタ側からサービスにアクセスすることで機能を実現するものである。もともとは Visual Studio Code のために開発されたものであるが、現在では多くのエディタがサポートしているため、提案手法を複数の開発環境に対して低コストで展開することが期待できる。

LSP では通知頻度を設定することができる。ここでは、ファイルが編集されるたびにイベントを発生させることで、言語サーバは一文字単位でインクリメンタルに編集内容を受け取る。受け取った編集履歴は、桑原らの関連研究 [4] で提案されている手法を参考に隣接する編集操作を統合する。統合したデータはエディタからの指示によって API サーバに送信される。

図 2 は送信される履歴の一部である。イベント発生時刻やファイル名、編集内容が記載された JSON フォーマットのデータとなっている。この図に示したデータのほか、ファイルの編集開始前の状態もあわせて送信することで、編集内容を再現できるようになっている。

2.2 細粒度編集履歴の分析部

分析部では、開発環境とは独立して、受け取ったデータ

を保存して解析する役割を担う。データの保存形式と分析ツールを疎結合にするために、保存のための仕組みと解析のための仕組みを分割する。エディタが収集した細粒度編集履歴は、HTTP 通信を通じて API サーバに送信される。API サーバは、受信したデータを DB サーバに格納し、またデータ分析のために取り出すことのできる API を提供する。細粒度編集履歴の分析は、この API が提供するデータを用いて行う。

現在の実装では、DB サーバには収集した細粒度編集履歴のサマリだけを保管し、内容はローカルストレージもしくは NextCloud 等のオンラインストレージ上で圧縮ファイルとして保持している。

分析ツールとして、ブラウザ上で細粒度編集履歴を再生するためのツールと、ある期間において終了時のコードを 100%としたときの進捗可視化ツールを試作した。現状は API 経由ではなく手作業で履歴ファイルを指定している。

3. おわりに

本稿では、特定のエディタに依存せず、他のツールと容易に連携することができる細粒度編集履歴収集プラットフォームを提案し、その構成について説明した。

現在は開発環境側のプラグインは Visual Studio Code 用のものが実装済みであり、今後他のエディタ用のプラグインの開発を行う予定である。また、細粒度編集履歴に対する既存の分析手法を本プラットフォーム上で実装し、エディタや言語が異なる環境での実験を行うことも考えている。

API の仕様や具体的な細粒度編集履歴の内容については、今後詳細な検討が必要があると考えている。例えば現在の実装では通信量と煩雑な情報の削減のために言語サーバ側での要約処理を行っているため、1 文字単位の編集履歴は開発者側から送信される段階ですでに残っておらず取得できない。また、エディタの操作についても記録を行っていないため、開発者の行動に基づくフィードバックをしたい場合など、どのような情報を記録するかについて検討し、API を設計する必要があると考えている。

謝辞 本研究は JSPS 科研費 JP18H04094, JP19K20239 の助成を受けたものです。

参考文献

- [1] 大森隆行, 林 晋平, 丸山勝久: 統合開発環境における細粒度な操作履歴の収集および応用に関する調査, コンピュータソフトウェア, Vol. 32, No. 1, pp. 60-80 (2015).
- [2] : StackOverflow Developer Survey 2019, <https://insights.stackoverflow.com/survey/2019>.
- [3] : Language Server Protocol, <https://microsoft.github.io/language-server-protocol/>.
- [4] 桑原寛明, 大森隆行: 編集操作履歴の再生における粗粒度な再生単位, コンピュータソフトウェア, Vol. 30, No. 4, pp. 61-66 (2013).