

# プログラミング演習における Electron を用いた進捗把握による学習支援

## Learning support for programming exercises by monitoring progress using Electron

大山 哲平<sup>\*1</sup>, 香川 考司<sup>\*2</sup>  
Teppei OYAMA<sup>\*1</sup>, Koji KAGAWA<sup>\*2</sup>  
<sup>\*1</sup>香川大学大学院創発科学研究科

<sup>\*1</sup>Graduate School of Science for Creative Emergence, Kagawa University,

<sup>\*2</sup>香川大学創造工学部

<sup>\*2</sup>Faculty of Engineering and Design, Kagawa University

Email: s22g354@kagawa-u.ac.jp

**あらまし**: 大学等におけるプログラミング演習において、教員が学習者の進捗をリアルタイムに把握することは難しい。Web ベースのシステムでは、学習者がファイル提出を行うまで進捗を把握できない。そのため教員が学習者に適切な指導を行うことは難しい。学習者の提出を待たずに学習者の進捗を把握するためにローカルファイルの監視を行い、変更履歴を収集する。また、導入や更新を簡単に行うため、Electron を用いたデスクトップアプリケーションを実装した。収集した進捗データをもとに、教員が学習者の変更履歴やファイル名、更新頻度を確認する Web ページを実装した。評価実験は学部生 4 名を対象としアンケートを実施したところ、導入が簡単なデスクトップアプリケーションによるローカルファイルの監視という目的は達成できた。一方で、進捗データの可視化による行き詰まりの発見には機能の不足が見られた。

**キーワード**: 学習支援システム, Electron, デスクトップアプリケーション, ファイル監視

### 1. はじめに

香川大学におけるプログラミング演習では、各自のノートパソコンを用いて演習を行う。演習を完了した場合、提出を LMS や Web ベースの学習支援システムにて演習ファイルの提出を行う。授業中わからない問題があった場合、挙手や学習支援システム等のチャットにて教員や TA に質問をする。教員が学習者の進捗を把握する手段は、演習ファイルが提出されるか、学習者側から質問がなされた場合に限られる。そのため、学習者の進捗の把握が難しく、演習中に教員側から学習者に対して適切な指導をリアルタイムに行うことは困難である。

現在、プログラミング演習における学習支援システムは学習者にとって導入が簡単な Web ベースシステムを中心に開発されている。Web ベースシステムでは、進捗の確認はソースコードの提出や、Web ベースのエディタを利用して行われてきた<sup>(1,2)</sup>。ソースコードの提出による進捗の把握では、学習者の提出を待つことになりリアルタイムな学習支援を行うことは難しい。また、Web ベースのエディタを用いると、学習者の使い慣れたエディタを使えない、コマンドプロンプトを用いたコンパイル等の操作の練習が行えないといった問題が発生する。学生の演習環境でサーバとして起動するログ収集器を用いる研究<sup>(3)</sup>では、エディタやコンパイラを用いる演習環境でログの収集が行える。しかし、各自のノートパソコンを用いる香川大学におけるプログラミング演習では、システムの導入や更新が難しい。

本研究では、Electron<sup>(4)</sup>を用いて作成したデスクトップアプリケーションによるローカルファイルの監

視を行う。ローカルファイルを監視することで、学習者が使い慣れないエディタを使うことなく、教員がリアルタイムに学習者の進捗を把握することができる。変更履歴や、変更頻度、変更時刻などの情報を収集することで、教員が学習者の進捗をリアルタイムに把握し、より適切な指導が可能になると考えられる。

### 2. Electron

Electron は Chromium と Node.js を用いたデスクトップアプリケーション開発用のフレームワークである。そのため、JavaScript や Html, CSS などの Web 技術を用いてデスクトップアプリケーションを開発することができる。Electron を用いることで、同一のコードで macOS, Windows, Linux 上で動くデスクトップアプリケーションを開発することができる。また、Node.js の機能や JavaScript のモジュールを用いることができ、開発が簡単に行える。アプリケーション配布用や自動アップデート用のモジュールも準備されている。

### 3. 目的

本研究では、学習者の進捗を把握するために次の要件を満たすシステムの開発を行う。

- デスクトップアプリケーションによる、ローカルファイルの監視を行うこと
- クロスプラットフォームであること
- 導入方法が簡単であること
- 収集したログから学習者の進捗がリアルタイムに確認できること

本システムは授業中における、学習者の進捗を確認することを目的としている。リアルタイムに学習者の進捗を確認するために、デスクトップアプリケーションによるローカルファイルの監視を行う。そのため、学習者は使い慣れたエディタを用いて演習をすることができる。デスクトップアプリケーションの開発には、Electronを用いる。Electronを用いることで、Windows, macOS, Linuxで動作するデスクトップアプリケーションを作成することが容易になる。Electronには、自動更新やインストーラの作成用のモジュールが備わっているため、学習者自身が導入や更新といった操作を行うことが簡単にできる。ローカルファイルの監視では、学習者が演習中のソースコードやファイル名、変更時刻を収集する。収集した進捗データはデータベースに送信され、教員が学習者の進捗を一覧や個別に確認できるWebページを作成する。学習者の進捗が把握できるように、変更履歴やファイルサイズの推移を用いてグラフを作成する。また、一覧で更新の少ない学習者をハイライトし確認できるようにする。

クポイントを設ける機能を追加する。ソースコードを構文解析し、学習者が特定のキーワードや構文を用いているかなどを判定する。また、ソースコードのコメントを除いた行数や変更内容の監視を追加する。変更内容は行の追加、削除、変更等である。ファイルサイズの監視よりも明確に学習者の進捗が確認できることが予想される。また、監視項目を増やすことで機械学習を用いた行き詰まりの検知への応用を検討している。

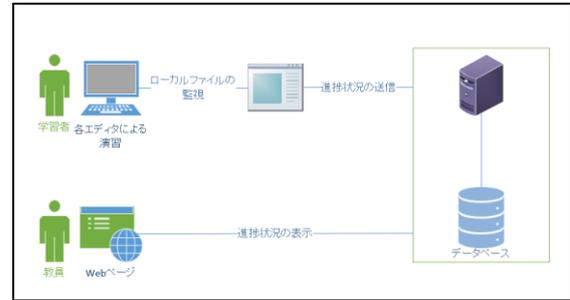


図1 システムの構成図

#### 4. システムの概要

システムの構成図を図1に示す。学習者の進捗を確認するデスクトップアプリケーションの開発にはElectronを用いる。学習者に自身の進捗情報が確認できるように図2のような表示を行う。変更頻度が少なくなっていることや、演習中のファイルサイズの推移、変更履歴を表示する。収集した学習者の進捗情報をサーバへ送る。サーバの開発にはNode.jsを用いる。集めた学習者の進捗情報を表示し、教員が学習者の進捗を把握できるようにする。変更頻度が下がった学習者やファイルサイズの推移が少ない学習者を強調表示する。各学習者の進捗の詳細を表示することで、各学習者の行き詰まりを確認する。行き詰まりの確認には、ファイルサイズの推移、最終更新からの時間、変更履歴等を用いる。変更履歴や更新頻度から行き詰まりの原因を把握し、適切な指導を行うことを目的としている。



図2 実行画面

#### 5. まとめ

本研究では、大学等におけるプログラミング演習において、教員が学習者の進捗をリアルタイムに把握することは難しいといった問題を解決するために、学習者の進捗把握のためElectronを用いたデスクトップアプリケーションによるローカルファイルの監視を行うシステムを実装した。教員が学習者の進捗を把握するために学習者の変更履歴やファイル名、更新頻度を表示する機能を実装した。評価実験は学部生4名を対象としアンケートを実施したところ、導入が簡単なデスクトップアプリケーションによるローカルファイルの監視という目的は達成できた。一方で、進捗データの可視化による行き詰まりの発見には機能の不足が見られた。

今後は、学習者の進捗を可視化するため、チェッ

#### 参考文献

- (1) 井垣宏, 齊藤俊, 井上亮文, 中村亮太, 楠本真二: “プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PVの提案”, 情報処理学会論文誌, Vol.54, No.1, pp.330-339 (2013)
- (2) 高橋功欣, 小島佑介, 北英彦: “プログラミング演習における指導のための受講者のコーディング状況の可視化”, JAEIS 第1回全国大会講演論文, (2011)
- (3) 小暮悟, 杉山匠, 野口靖浩, 小西達裕, 伊東幸宏: “プログラミング演習における演習履歴自動収集環境とそれを利用した授業改善の枠組み”, 2017 人工知能学会全国大会, 第31回, 1F1-OS-26a2 (2017)
- (4) Electron: “<https://www.electronjs.org/>”, 最終アクセス 2022年5月20日。