

B-016

## 音声を利用したソースコード理解支援ツールの作成

Development of a tool which uses developer's voice for source code

吉原駿\*<sup>1</sup>                      深澤良彰\*<sup>1</sup>  
Shun Yoshiwara      Yoshiwaki Fukazawa

\*<sup>1</sup> 早稲田大学基幹理工学部情報理工学科  
Department of Computer Science and Engineering, Waseda University

## 1.はじめに

近年、IoT 技術や AI 発展などが話題になっている。これらの技術の実現には、技術者によるプログラミングが必要不可欠である。プログラミングではソースコードを作成するだけではなく、他人のソースコードを保守、メンテナンス等のために閲覧することはよくある。一般的に他人が作成したソースコードを読み、それを完全に理解するのは決して容易ではない。この問題を解決することはプログラミングにおける技術的進歩につながる。

Christopher Oezbek らはこの問題に対する研究を行なっていて、ソースコードとそれを説明するためのツアー構造の文書があればプログラム理解が容易になることを示している[1]。

本研究ではこの問題を解決するために、音声を利用したプログラム理解の容易化手法を提案する。プログラマーが考えていることを音声データとして残し、その音声とソースコードを紐づけることでこの問題を解決することができるのではないかと考える。

## 2.本提案手法の特徴

本提案手法の特徴として、まず音声を利用することでコメントや文書などと違い、ソースコードを見ながらそのソースコードの情報を知ることができる点がある。コメントなどは読むために一度ソースコードから目を離す必要があるが、音声は耳から情報を得るため、目を離す必要がない。

また音声ファイルを1つのソースコードに対して関数やループなどのまとまりごとに細分化して、複数紐づけするため、ソースコード読者は必要な部分のみ解説を聴くことができる。また音声ファイルなので同じ部分を何度も再生することができるのも長所といえる。

これらに加えて、音声解説されているソースコードの部分を視覚的に見やすくすることで、理解しようとしている部分を見失わず集中して閲覧することができる。

## 3.提案手法

ソースコード理解を容易化するツールを作成するにあたって、以下の5点の要素が必要になってくると考える。

- (1) ソースコードを説明するもの  
(コメントや仕様書など)
- (2) (1)で用意したものとソースコードを対応付ける仕

組み

- (3) 任意のタイミングで任意の場所を理解できるような仕組み
  - (4) 局所的にも、大局的にも範囲を指定できるような仕組み
  - (5) ソースコード読者が理解しようとしている部位とそれ以外の部位を視覚的に区別する仕組み
- これら5点の要素を本論文では、「理解支援に必要な要素」として取り扱う。

以上の理解支援に必要な要素を踏まえた本研究の提案手法について、ソースコード作成者とソースコード読者を分けて説明する。

## 3.1.ソースコード作成者

プログラム作成者はソースコードを作成する際に、記述している部分について何をしている部分なのか、どういう意図で記述しているのかななどの思考を音声データとして記録する。この音声データは理解支援に必要な要素(1)に該当する。

音声ファイルについては1つのソースコードに対して1つではなく、関数やループなどの1まとまりで1つとする。1つの音声ファイルの長さを短くすることで理解支援に必要な要素(4)の局所的な範囲の指定に役立つ。

その後作成したソースコードと録音してできた音声のファイルを結び付ける。この際に音声ファイルの内容とソースコードの内容が対応する様な紐づけを行う。

## 3.2.ソースコード読者

ソースコード読者は、ソースコードと音声ファイルの紐づけが行われたものを受け取る。

音声ファイルを再生するためのユーザーインターフェースが用意されていて、読者はそのインターフェースを通してソースコード作成者の考えを聴く。この際に読者は、全体を聴くことも、考えを聴きたい部分だけを聴くことができるようにする。また聴くソースコードの範囲も読者が指定できるようにして、自由性を持たせる。これは理解支援に必要な要素の(3),(4)に該当する。

## 4.ツールの作成

本研究では提案手法を実現させるためのツールの作成も行った。ツールの作成をする際に開発ツールとしてUnityを使用した。Unityはもともとゲーム作成向けの開発環境であり、UIやAPIがあらかじめ多

く存在していて開発しやすい環境であると判断したため本研究で採用した。開発した2つのツールを説明する。

#### 4.1. 録音ツールの作成

理解支援に必要な要素(1)に本研究ではソースコード作成者の考えや意図をまとめた音声データを採用した。そのためプログラム作成者が音声データを残すための録音するツールを作成した。

録音ツールは、録音の開始、終了とデモ視聴、ローカル上に保存するという機能を持っている。保存形式は wav 形式を利用している。

これによって理解支援に必要な要素(1)に該当する要素を実装することができた。

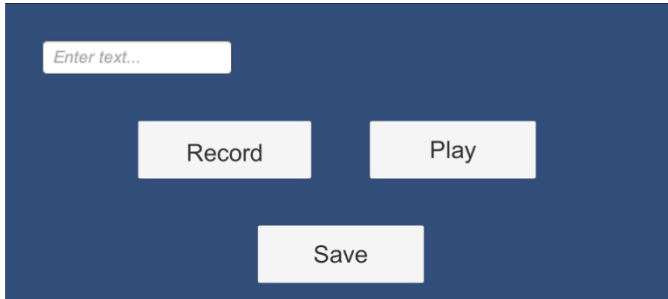


図 1. 作成した録音ツール

#### 4.2. ソースコードと音声ファイルを紐づけるツール

ソースコード読者が、音声ファイルを聴きながらソースコードを読むことができるようにこれらの紐づけができるツールを作成した。

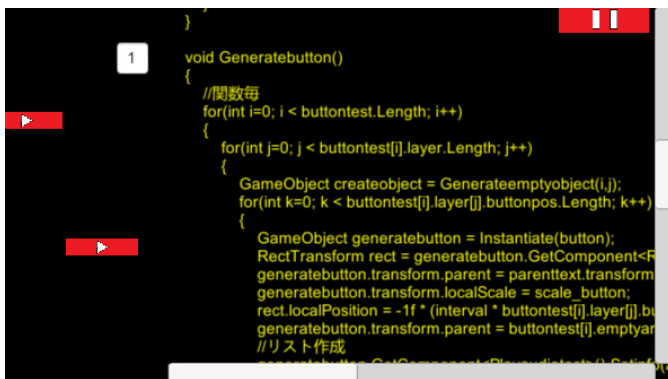


図 2 作成したソースコードと音声ファイル紐づけツール

図 2 に作成したツールを使用している様子を記載している。

黄色い文字で表示されている文字列は、読み込まれたソースコードでこれをソースコード読者が理解しようとする状況を想定している。

理解支援に必要な要素(2)を実現するために、音声ファイルとボタンにそれぞれ対応させる行数を設定する。ボタンが押されたときにそのボタンに設定された行数が登録されている音声ファイルを再生する。複数の音声ファイルがある場合には、設定された行数が若い順番に再生する。例として、1 から 5 行目を設定したボタンと、1 から 2 行目、3 から 5 行目を登録した音声ファイルが存在するとしたとき、ボタンを押すと 1 から 2 行目に対応した音声ファイルが再生され、そのあとに 3 から 5 行目に対応した音

声ファイルが再生される。

様々な行に対応させたボタンを複数配置できることによって理解支援に必要な要素の(3)と(4)を実現させることができた。

理解支援に必要な要素(5)は、音声ファイルが再生されたときに、その音声ファイルに設定した行の部分の文字の色を変更するようにして実現した。

視覚的に見やすくする手法として、文字の色を変更する手法以外に文字のサイズを変更、また文字を斜体にするなどの手法も考えられる。どの手法が適しているかは今後の有用性の検証の結果を踏まえて適宜変更する。

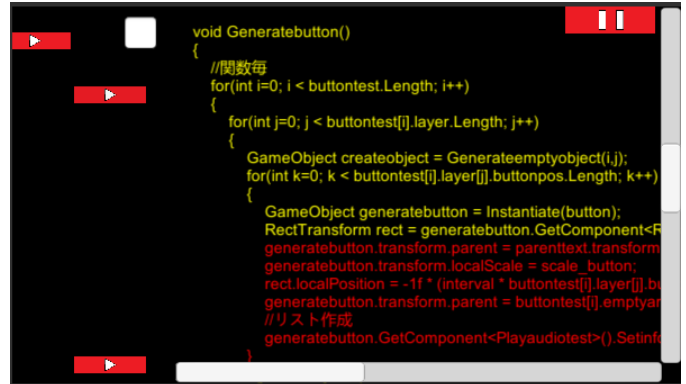


図 3. 音声ファイルに対応した部分のソースコードの色が変更されている様子

#### 5. おわりに

本研究ではソースコード理解を容易化するために、音声に着目した支援ツールを使用する手法を提案し、そのツールの作成を行った。実現したこととして以下の3点が挙げられる。

- ・ 音声を録音する仕組み
- ・ 音声ファイルとソースコードを対応付けする仕組み
- ・ 音声ファイルを再生する仕組み

当初の目的を果たすために必要な機能は実装したが、このツールの有用性については検証できていない。

また本ツールは Unity 上でしか動作ができない問題もある。ソースコードは基本的にテキストエディタを用いて作成されるので、テキストエディタ上において本研究で作成したツールと同様の機能を実現できないか検討する必要がある。

今後の課題をまとめると次の2点である。

- ・ 作成したツールの有用性の検証
- ・ テキストエディタ上で本研究にて作成したツールと同様の機能を実現させる

#### 6. 参考文献

- [1] Christopher Oezbek, Lutz Prechelt. "JTourBus: Simplifying Program Understanding by Documentation that Provides Tours Through the Source Code". 2007 IEEE International Conference on Software Maintenance, 2-5 Oct. 2007. <https://ieeexplore.ieee.org/abstract/document/4362619>, (参照 2021-06-07)