

# ソースコードの注釈をプログラミングの知見として共有する ソフトウェア

## — コーディング過程から学ぶプログラミング —

### 1. 背景

これまでプログラミングを学ぶ際には書籍やネット記事に記載されているソースコードを読んで理解するという方法が採られていた。しかしこの方法はソースコードの行間から意図を理解する能力が必要であり、それには経験と知識を必要とする。完成されたソースコードから知見を得ることができるのは経験者であって、初学者には負担が大きい。つまり完成したソースコードから学ぶ、従来のプログラミング学習は初学者に適していない。そこで完成されたソースコードを解説する従来の「結果の解説」ではなく、ソースコードが完成に向かっていく過程を解説する「過程の解説」を実現することを考えた。

### 2. 目的

本プロジェクトの目的は初学者にとってのプログラミング学習の負担を減らすことである。ソースコードがどのような順序で作られていったか、開発者が何を考えてソースコードを書いたのかといった情報をデータ化し、過程を可視化する。経験者が持つ知見をソースコードに関連付けてデータとして保存する。これによって知見から新しい知識を得ながら、可視化された過程を真似してソースコードを書くことができるようになる。経験と知識に依存した「結果の解説」ではなく過程を真似する「過程の解説」を提供することでプログラミング学習の負担を減らす。

### 3. 開発の内容

本プロジェクトでは、プログラミングの効果的な学習方法として「過程の解説」を提案し、コーディングの過程の記録と知見の注釈を行うシステム“Disconomy”を開発した（図1）。

```
4  cgi = CGI.new
5  YOYFF
6
7  ファイル操作関数のgetsで読み取った文字列の末尾には改行文字が
8  付いているのでchomp関数で削除すべきである
9  ここは文字列の改行文字の有無なので直接的なエラーには関係ないかもしれな
10 いが
11
12  怪しい処理は早いうちに直しておくのが良い
13  ...while line = $io.gets
14  ..  c += 1
15  ..  if c = 1 then
16  ..  ..  puts "<h2>#{line}</h2>"
17  ..  else
18  ..  ..  puts "<p><input type='checkbox' name=gumi value=#{line}>#{line}</p>"
19  ..  end
20  ..  io.flock(File::LOCK_UN)
21  ..  end
22  print <<EOF
```

図1. ソースコードに知見を注釈付けることができる。

Disconomyでは様々な形式でソースコードに知見を注釈付けることができる。プレーンテキストによる注釈だけではなくMarkdown形式のテキストによる注釈（図2）や画像による注釈（図3）も可能である。

```
1 void merge(int *a, int *tmp, int l, int r, int mid);
2 void merge_sort(int a[], int l, int r)
3 {
4     int tmp[r - l + 1];
5     int mid
6     YOYFF
7
8     Disconomyの注釈ではMarkdown記法が使えます
9     Boldにしたり、Italicにすることができます
10    これによって用語の強調が可能となり、学習者の理解を進めることができます
11
12    merge_sort(a, l, mid);
13    merge_sort(a, mid + 1, r);
14
15    merge(a, tmp, l, r, mid);
16
17    return;
18 }
```

図2. Markdown形式で注釈付けることができる。

```
1 void merge(int *a, int *tmp, int l, int r, int mid);
2 void merge_sort(int a[], int l, int r)
3 {
4     int tmp[r - l + 1];
5     int l
6     YOYFF
7
8     また言葉で説明するより画像で説明した方がいいときもあります
9
10
11
12
13
14
15
16
17    return;
18 }
```

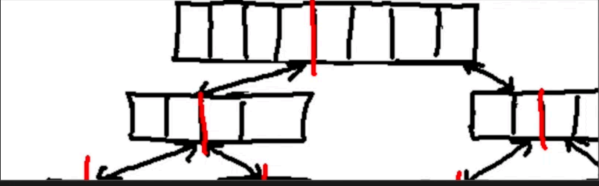


図3. 画像で注釈付けることができる。

Disconomyでは入力された注釈をJSONファイルに保存している。JSONファイルにはどのファイルの、どの場所に対して、どのような注釈を付けたのかを保存している。そのため、Disconomyで作成したJSONファイルを共有することで第三者が作成した注釈を見ることが可能である。これによって第三者への知見の共有が可能となる。

さらにDisconomyではコーディングの過程を記録しており、ソースコードが完成に至るまで道筋を見ることができる（図4.1, 4.2）。

```

1 |#!/usr/bin/env ruby
2 |# encoding: utf-8
3 |
4 |YOYFF
5 |CGIのインスタンスを初期化
6 |cgi = CGI.new

```

図4.1. 過程を記録することができる。

```

1 |#!/usr/bin/env ruby
2 |# encoding: utf-8
3 |
4 |require 'cgi'
5 |YOYFF
6 |ヘッダー情報を出力
7 |
8 |print cgi.header("text/html;charset=utf-8")

```

図4.2. 過程に注釈付けることができる。

コーディングの過程とそれぞれの過程につけられた注釈をまとめて「注釈群」として保存する。コーディングの過程に一意的識別子を与え、注釈群を管理するJSONファイルでコーディング過程の順序を識別子によって管理している。これによってコーディングの過程を先に進めること（図5の赤矢印）やコーディングの過程を前に戻すこと（図5の青矢印）が可能になる。

プログラムの意図や目的が理解できている状態まで過程を巻き戻して、その状態から過程を先送りすることで、理解できている箇所とそうでない箇所を明確にして、理解できていない箇所を重点的に学ぶことが出来る。これは従来の「結果の解説」では実現できない学習方法である。

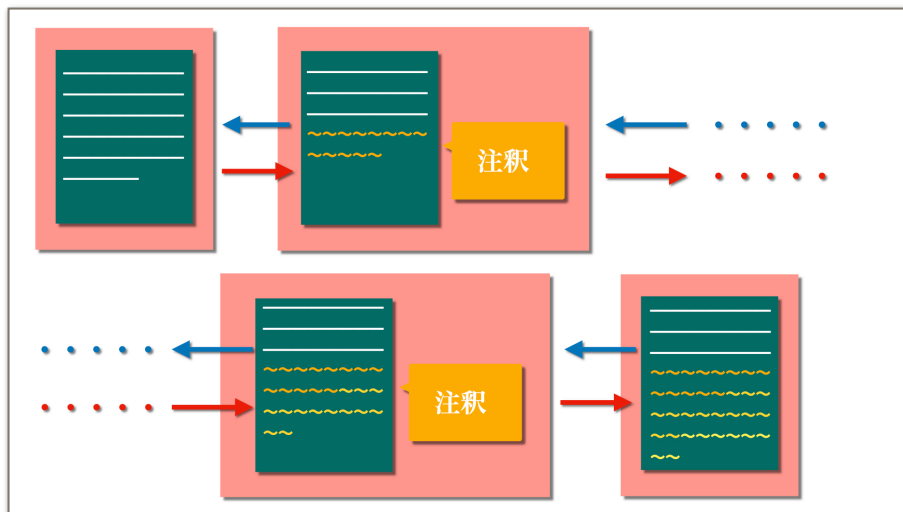


図5. 過程の先送りと巻き戻し

DisconomyはJSONファイルの書き出し、読み込みを行うVSCode拡張機能であり、ファイルの書き出し、読み出しという単純な機能の組み合わせで「過程の記

録」「ソースコードへの注釈」を実現した。特定のプラットフォームやOSに依存してないため、他のソースコードエディタへの移植を容易に行うことが出来る。

Disconomyではショートカットキーによる注釈の作成と過程の記録が可能である。コーディングと並行して注釈の作成と過程の記録が出来るため、ユーザが負担を感じることなく、注釈の作成と過程の記録が出来る。

#### **4. 従来の技術（または機能）との相違**

Disconomyでは完成したソースコードの解説だけでなく、コーディング過程の解説ができる。従来のソースコード解説ツールは完成したソースコードの解説、つまり「結果の解説」しか行わない。Disconomyでは「過程の解説」を行い、初学者に負担の少ないプログラミングの学習方法を提供する。

#### **5. 期待される効果**

Disconomyが提供する「過程の解説」によってプログラミング学習の敷居が下がり、プログラミングを習得する人が増えることが見込まれる。これによって不足しているIT人材の確保に繋がる。また、プログラミング経験者の有益な知見を共有することでプログラミング中上級者のスキルアップに繋がり、高度なIT人材の育成は可能となる。

#### **6. 普及（または活用）の見通し**

Disconomyは高等学校や大学におけるプログラミング教育で利用することを想定している。クリエイターが在籍する筑波大学の学生生活動支援プロジェクトを利用してDisconomyを大学の授業で導入することを目指しており、最終的には全国の情報系の学部 に在籍していると言われていた約2.8万人の学生に使用してもらいたいと考えている。

#### **7. クリエータ名（所属）**

藤永 弥太郎（筑波大学情報学群情報メディア創成学類）