

# プログラミング演習における個別指導のための コーディング状況把握方法の提案 活動度の観点から

2015SE038 川出涼雅

指導教員：蜂巢吉成

## 1 はじめに

大学におけるプログラミング演習では、学習者数十名、教員 1 名 TA 少数名という形式で授業が行われる事が多く、指導者（以下、教員と TA を指導者と呼ぶ）は学習者に対し効率的な指導が求められる。学習者の質疑に答えたり、指導者が巡回して、ソースプログラムを読み、行き詰まった学習者を指導したりする。これでは、限られた時間内で指導者が多くの学習者を指導するには限界がある。

本研究では、編集作業が滞っている学習者を特定するために、活動度の観点から学習者のコーディング状況を把握する方法を提案する。活動度とは学習者がどれくらい作業しているかを表したものである。単位時間あたりに増加したソースコード文字数及びコンパイルエラー継続時間の 2 つを評価基準として用いる。単位時間あたりの増加文字数が少なければ、学習者は考え込んでおり手が止まっていると考えられるので、コーディング過程でつまづいていると言える。コンパイルエラーは記述ミスが原因と考えられるので、エラー継続時間が長い学習者は指導が必要となる。しかし、活動度としてみる増加文字数の計測時間が短い場合には、学習者が少し考え、その間手が止まっているだけで活動度が低いと捉えられてしまうことが考えられる。また、活動度としてみる増加文字数の計測時間が長い場合には、直近に編集していなくても活動度が高いと捉えられてしまうことが考えられる。そのために、活動度における適切な時間設定と重み付けを行う。本研究では、実際の演習データから活動度の適切な重み付けについて考察する。

## 2 関連研究

文献 [1] では、ソースコードの行数など 4 種の観点から相対的にコーディング遅延を把握する。文献 [3] では、コンパイル回数など指導者が各種履歴を組み合わせ、それらより遅延している学習者を把握する。文献 [1] では相対的に遅延する学習者を特定する点、文献 [3] では指導すべき学習者の優先順位をつけられない点で、本研究とは異なる。

## 3 プログラミング演習における活動度の提案

編集作業が滞っている学習者を特定するために、学習者の作業量を表す、活動度を定義する。活動度は 0 から 100 とし、100 に近いほど活動しているものとする。

前提として、文献 [2] の WebIDE を用いてソースコード及びコンパイル情報を収集し、ソースコードは 1 分ごとに、コンパイル情報はコンパイル時にそれぞれ保存する。

### 3.1 ソースコード文字数の増加量の分析

学習者ソースコードに対し、毎分ごとに増加した文字数に着目した分析を行う。学習者ごとの単位時間あたりに変化したソースコード文字数を計測する。学習者  $s$  の時刻  $t$  における活動度  $act(s, t)$  の定義を次に示す。

$$act(s, t) = 100 \sum_{i=1}^N w_i \frac{x'_{s,t-i+1}}{B}$$

$$\sum_{i=1}^N w_i = 1, \quad x'_{s,t} = \begin{cases} \min(x_{s,t}, B) & (x_{s,t} \geq 0) \\ 0 & (\text{otherwise}) \end{cases}$$

$x_{s,t}$ : 学習者  $s$  の時刻  $t$  から  $(t-1)$  間で変化したソースコードバイト数

$B$ : 1 分間あたりに変化する文字数の基準値 (バイト数)

$w_i$ :  $i$  分前から  $(i-1)$  分前の重み

$N$ : 活動度として計測する時間 (分)

$x_{s,t}$  の値が負の場合には、ソースコードの編集が進んでいないものとし、 $x'_{s,t}$  の値を 0 とする。

### 3.2 コンパイルエラーの分析

コンパイルエラー継続時間及びエラーの種類に着目した分析を行う。エラー継続時間については、エラー発生から修正されるまでに要した時間とする。エラーの種類ではエラー文を観測する。エラー文には特定文字の欠落等、エラーの原因が表記されることが多い。エラー継続時間が長い程、その箇所にて学習者の記述に誤りがあり、指導が必要である。

## 4 実験

文献 [2] の演習データを用い、次の実験を行った。

予備実験 変化したソースコード文字数の平均値及び各問の解答時間の平均値を求める。

実験 1 活動度の妥当な計測時間  $N$  を確認する。

実験 2 活動度への重み付けにより、指導が必要な学習者を見つけやすくなるかを確認する。

実験 3 コンパイルエラー継続時間及びその種類が指導の材料となるかを確認する。

演習問題は 5 問あり、問 2 は実数のべき乗を計算する関数を作成するものである。実験 1, 2 では予備実験より  $B=5.8$  を用いる。実験 1 では、 $N=1$  のとき  $w_1 = 1, N=5$

のとき  $w_i = 0.2 (i = 1, 2, \dots, 5)$  を用い, 実験 2 では実験 1 より  $N=5$  を用いる. 活動度が 30 以下の場合を活動度が低いものとする. 実験 2 における重み付けとして, より直近の情報が学生への影響が大きいと考え, 次を用いる.

重み A 各 0.2 に重み付け.

重み B 直近から 0.3, 0.25, 0.2, 0.15, 0.1 に重み付け.

重み C 直近から 0.5, 0.3, 0.1, 0.075, 0.025 に重み付け.

#### 4.1 結果

予備実験では, ソースコード文字数の変化の平均値は 5.8 であり, 標準偏差は 16.2 であった. また, 学生 11 名の各問の解答時間の平均値は問 1 から順に 8.4 分, 8.6 分, 7 分, 13 分, 13.2 分であった.

実験 1 では, 図 1, 2 に各学生の問 2 の解答時間を示す. 垂直軸は学生である. 水平軸は問 2 を解き始めてからの経過時刻  $t$  である. 棒グラフの長さは解答時間であり, 活動度が 30 以下となった時間から赤色とする.

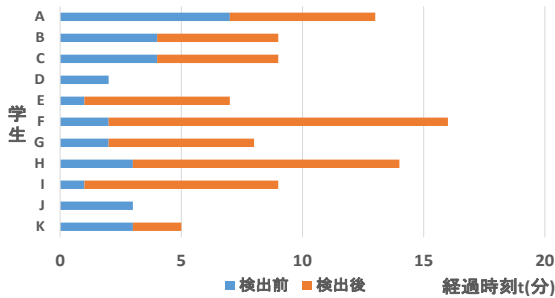


図 1 実験 1: 学生の問 2 の解答時間 (N=1)

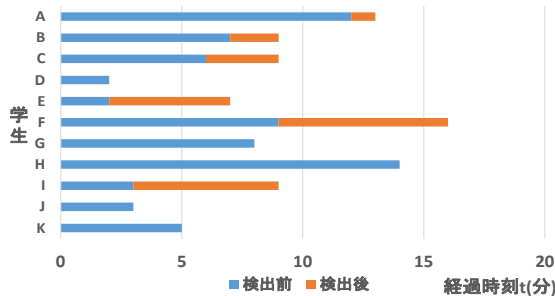


図 2 実験 1: 学生の問 2 の解答時間 (N=5)

実験 2 では, 図 3 に学生 A の解き始めから 20 分経過時点までの活動度への重み付けを示す. 垂直軸は活動度, 水平軸は経過時刻  $t$  である.

実験 3 では, 計 79 件のコンパイルエラーが発生し, 内 59 件が発生 1 分以内に修正された. 継続時間が 3 分以上であったものとして, 演算子の不適切な利用がある. 実際のエラー文を次に示す. エラーの原因は `int *` の `min` の前に `*` が記述されていないことである.

```
1 invalid operands to binary expression ('int*' and
   'int')      s=s-(min*60);
```

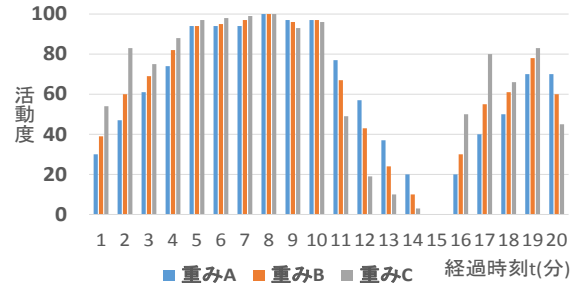


図 3 実験 2: 学生 A の活動度への重み付け

## 5 考察

予備実験より, 問 1 で模範解答に至るまでの文字数は 55 文字である. 変化した文字数の平均値 5.8 を参考にすると約 10 分で到達する計算となる. 問題数と演習時間から,  $B$  にて変化した文字数の平均値を用いることは妥当である.

実験 1 より,  $N=5$  では図 2 の赤色となる時刻にて指導することで, 赤色の部分の長さが短縮し, 解答時間を短くすることができる可能性がある.  $N=1$  では検出する学生数が多く, 解き始めに集中している. これでは指導すべき学習者を判断できないので,  $N$  は 5 分の方が妥当である.

実験 2 では活動度が 30 以下となる時刻に大きな差は見られなかった (図 3). 活動度への重み付けにより指導が必要な学習者を特定しやすくなるとは言えない.

実験 3 より, 演算子の不適切な利用はポイントに関する間で発生し, エラー継続時間は 3 分以上であった. 比較的修正難易度が高いと言える. このエラー発生時には指導が必要であり, エラーの種類は指導の材料となりうる.

## 6 おわりに

プログラミング演習における活動度の観点から, 学習者のコーディング状況を把握する方法を提案した. 単位時間あたりに増加したソースコード文字数及びコンパイルエラー継続時間に着目し, 実験を行い, 個別指導が必要な学習者を把握できることを確認した. 様々な演習での評価が今後の課題である.

## 参考文献

- [1] 井垣宏, 井上亮文, 齊藤俊, 山田誠: プログラミング演習における学生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 (2014).
- [2] 石元慎太郎, 蜂巢吉成, 吉田敦, 桑原寛明, 阿草清滋: プログラミング演習における構文要素の種類毎のビューによるコーディング状況把握方法の提案, 情報処理学会情報教育シンポジウム, 8 pages (2018).
- [3] 藤原理也, 田口浩, 島田幸廣, 高田秀志, 島川博光: ストリームデータによる学習者のプログラミング状況把握, 電子情報通信学会第 18 回データ工学ワークショップ, pp. 1-6 (2007).