

# 入学前教育 e ラーニングのプログラミング課題における チューター支援環境の構築

小山田圭吾<sup>†</sup> 市川尚<sup>†</sup> 富澤浩樹<sup>†</sup> 阿部昭博<sup>†</sup> 高木正則<sup>†</sup>

岩手県立大学大学院ソフトウェア情報学研究科<sup>†</sup>

## 1. はじめに

本学部では入試の早期合格者に対して e ラーニングによる入学前教育を実施している。プログラミングの課題も出題しているが、入学前教育参加者のプログラミング経験にはばらつきが見られる。そこで、本研究では、受講対象者である高校生をチューターと学習者に分けてグループを組ませ、学習者が作成したプログラムに対してチューターが確認してフィードバックを行うためのオンライン上のプログラミング学習環境を構築してきた<sup>1)</sup>。2018年度の実践の結果、チューターの確認の際に学習者のプログラミングの作業履歴を提示することはチューターから好印象であったが、チューターが不十分な確認をしていること、学習者の作業回数が多いとチューターの確認に時間がかかってしまうことが分かった。

本稿では、2018年度の実践結果を踏まえ、学習環境の改善を行い、2019年度の入学前教育に導入した結果について述べる。

## 2. 学習環境

### 2.1. 学習環境の概要

学習環境の構成を図1に示す。学習環境は、学習管理システムである Moodle と、それと連携する著者が開発したシステムから構成される。システムは、学習者がビジュアル型言語でプログラミングの課題を作成し提出するまでの履歴を記録し、チューターに履歴を提示することで、学習者へのフィードバックを支援する。図2はチューターのチェック画面であり、画面左は学習者が作成したプログラム、左下に履歴を閲覧するスライダー、左上が実行結果、右下がチェック項目となる。

### 2.2. システムの改善

本研究では、2018年度の実践結果を踏まえ、いくつか改善を行った。まず、学習者の作業履歴をチューターが確認する負担を減らすために、作業履歴を短縮して表示するようにした。上田らの研究<sup>2)</sup>では、レーベンシュタイン距離を用いて2つのソースコードの類似性を求めていた。本システムではそれを参考に、エディタ(プログラム)の状態を記録しているXMLと次の状態のXMLがどれほど異なっているかをレーベンシュタイン距離で算出して、0.0~1.0で

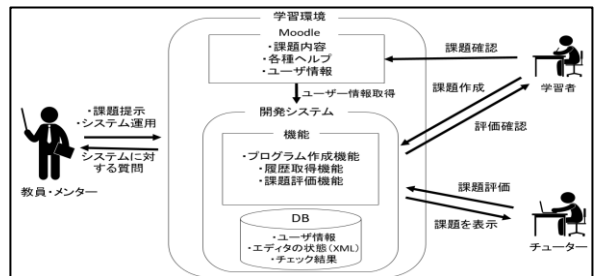


図1 学習環境の構成

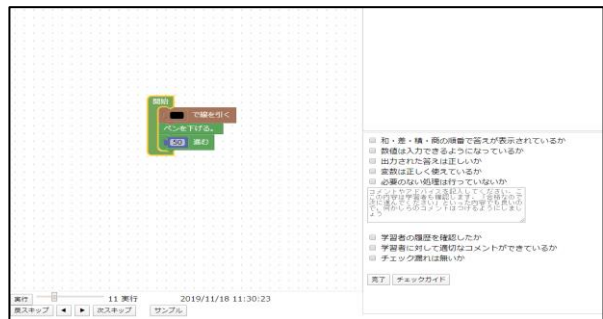


図2 チェック画面

表示するように正規化(1.0に近いほど類似)し、類似性が0.5を下回ったとき、特定の作業(課題を実行したとき、全消去する直前、提出)が行われたときのエディタの状態を中心にチューターが閲覧するように改善した。

次に不十分な確認を失くしてフィードバックの質を向上させることを目的に、プログラムの出力結果に関する5つのチェック項目に加えて、3つの項目(学習者の履歴を確認したか、学習者に対して適切なコメントができていないか、チェック漏れはないか)を追加した。さらに、チェックの不備をなくすために、チェック項目の内容を説明するガイドを提示した。また、チューターのチェック行動を確認するため、チューターの作業履歴を取得するようにした。

## 3. 2019年度の実践

### 3.1. 実践の概要

A0 入試合格者21名に事前テストとアンケートを実施し、それを基にチューターを選抜した。課題は11月11日~12月13日の期間に学習者がチェック項目を満たすまで提出とチェックを繰り返すという手順で取り組ませた。課題終了後、事後テストとアンケートを実施した。課題は制御構造の基本となる順次、分岐、反復をそれぞれ課題1~3、それらを組み合わせた課題4を用意し、それぞれの課題に対する説明を補足した。テストは昨年度と同じ内容であ

Development of e-learning environment to support tutors for programming tasks in preparatory education  
Keigo Oyamada<sup>†</sup>, Hisashi Ichikawa<sup>†</sup>, Hiroki Tomizawa<sup>†</sup>, Akihiro Abe<sup>†</sup>, Masanori Takagi<sup>†</sup>  
<sup>†</sup>Graduate School of Software and Information Science, Iwate Prefectural University

る。本実践は、学習者 16 名、チューター 5 名で進めた。チューターには、学習者に直接答えを教えないなどの注意を促すとともに、合格不合格に関わらず必ずフィードバックのコメントを付けてもらうようにした。チューターのチェック後に、メンターはチェックが妥当であったかの確認をしている。

### 3.2. チューターによる確認の結果

学習者 16 名全員がすべての課題を提出し、すべての課題について、1 回目不合格であっても再提出を経て、全員が合格に至った。また、メンターの最終確認時に、チューターの確認の妥当性を確認したところ、約 95.3%が妥当に行われていた。不備のある確認は、課題 1~3 でそれぞれ 1 件ずつ見られた。

次にチューターの作業を調査した。2 件不備があったチューターの作業は、実行結果のみで課題の評価を行っているような傾向が見られた。もう 1 人は履歴を細かく確認していたが不備が見られた。これらはガイドを読んでいれば回避できたものであり、ガイドを見ていない可能性もある。他にも、チューター全員が最低 1 回は履歴の確認をしており、担当した学習者全員の作業履歴を確認しているチューターもいれば、間違ったプログラムを提出したときのみ履歴を確認するという特徴も確認できた。

### 3.3. フィードバック時に付与されたコメント

チューターによるチェックは全部で 101 回行われた。特に不適切なコメント（同じ内容を繰り返す、課題内容をそのまま伝えるなど）は見られなかった。チューターによっては合格の場合でも、どのような点が良かったか、どう改善すればもっと見やすくなるかなど、補足のコメントをしていた。作業履歴に着目したコメントは 5 件であった。実際に付与されたコメントの例を以下に示す。

合格です！しっかりと動作することを確認しました。また、履歴を確認してみました。が、試行錯誤の跡が見て取れたのが良いと思いました。答えを別の変数に代入するというやり方は今回のような簡単なプログラムでは使うと煩雑になってしましますが、もっと複雑なプログラムを組む時には大切になってくるので、覚えておくの良いかもしれません！

和がうまく出ない理由は、変数の「型」にあります。数値型と文字列型があり、どちらにするかで「+」演算の意味も変わってくるようです。ですが、数値型に変えてもこの課題は終われません...。今回の課題は「実行時に値を入力できる」必要があります。何か使えそうなブロックはありませんか？ どうしても無理なら、課題1の説明をもう一度見てきてもいいかもしれませんよ。

### 3.4. アンケート結果

チューターを対象にアンケートを行い、5 名全員から回答があった。チューターとしての活動が良い経験になったかという質問には、自分には思いつかなかった考え方が分かったなどの理由から 5 名全員が肯定的な回答であった。作業履歴がチェックするに当たって役に立ったかを質問したところ、履歴を確認した 4 名がそう思うと回答していた。自由記述には、どんな過程で正解まで辿り着いたのか、どこで迷ったのかが分かったという記述が見られた。また、履歴の短縮表示とチェックガイドは役立ったかを質問したところ、履歴の短縮は 4 名、チェックガイドは 5 名が役立ったと回答していた。

### 3.5. 2018 年度の結果との比較

昨年度の AO 入試合格者が行った確認と比較した

表 1 2018 年度と 2019 年度の比較

	合格者	未チェック	未完成	最終確認
課題1	16(10)	0(0)	0(0)	15(10)
課題2	16(8)	0(1)	0(1)	15(8)
課題3	16(9)	0(1)	0(0)	15(9)
課題4	16(6)	0(2)	0(2)	16(6)

※()内は2018年度の結果

結果を表 1 に示す。昨年度は学習者 10 名、チューター 10 名で行っている。2018 年度は各課題で数名の学習者が、未チェックまたは未完成が原因で合格とならないまま課題が終了していた。それに対して、2019 年度ではすべての学習者が合格となるまで課題に取り組んでおり、未チェックと未完成の学習者は見られなかった。付与されたコメントの内容に関しては、2018 年度は計 11 件の不適切な内容が見られたが、2019 年度は不適切な内容は見られなかった。また、合格している学習者へのコメントも、2019 年度のチューターにはどのような点が良かったのかなどを記述している者もおり、質が向上していた。

## 4. 考察

実践の結果、不合格であった学習者も最終的には合格に至っており、チューターによる確認が学習者のプログラム改善につながっていた。2018 年度よりもチューターが適切な指導を行っていたことから、チェックガイドの追加とコメントの必須化の効果があったと考えられる。丁寧なコメントも見られた。学習者の作業履歴については、チューターは最低 1 回確認しており、アンケート結果からもチューターに好評なことが分かった。一方で、作業履歴に着目したコメントは 5 件に留まっていた。アンケートでは、学習者がどこで躓いたのか分かったという記述も見られたため、作業履歴を活用して更に効果的なフィードバックができる可能性があると考えられる。

## 5. おわりに

本稿では、システムを改善して導入し、得られた結果について述べた。評価の結果、昨年度のシステムよりもチューターの適切な指導の支援ができていたことが示唆された。今後は、履歴をより効果的に扱う支援方法を考えることが必要である。

## 参考文献

- 1) 小山田圭吾ほか：オンライン上での相互チェックを取り入れた入学前教育におけるプログラミング学習環境の開発，第 80 回全国大会講演論文集，pp.617-618 (2018)
- 2) 上田和志ほか：類似性に基づくレポート剽窃の検出ツールのプログラミング課題への適用，情報処理学会研究報告，Vol.2010-CE-107，No.9 (2010)