

ペアプログラミング利用時の学習状況把握支援

海野 芽美 伊藤 恵

近年、プログラミング導入教育においてペアプログラミングを用いることが増えている。プログラミング導入教育では、演習形式の講義が行われることが一般的である。そして、演習形式の講義では講師のサポート役として TA が導入されることが多い。しかし、学習者の数に対して少量のメンターが対応するために、学習者の状況を把握して支援することが困難な場合もある。そこで、本研究では、TA や講師などのメンターが学習者の状況を理解しやすくして、メンターによる効率的な学習支援ができるようにすることを目的とし、ペアプログラミングを使った演習授業における学習状況把握支援システムを提案する。本研究で提案するシステムでは、学習者のつまづきを検知・分類して、分類に応じた支援へ誘導することを目指す。本稿ではシステム提案のためのつまづき検知方法を提案する。

1 はじめに

プログラミング導入教育では、演習形式の講義が行われることが一般的である。本研究における演習形式の講義とは、講義中に演習課題が出され、学習者が自力で課題を解く授業形式のことである。

演習形式の講義では、講師はメンターとして学習者からの質問に対応すること、進捗が遅れている学習者を発見し支援をすることのような、学習者の状況に合わせた支援を行わなければならない。しかし、講師だけでこのような支援をすることが難しいため、追加のメンターとしてティーチングアシスタント (以後、TA) が導入されることが多い。

ただ TA を追加しても、学習者の数に対して少数のメンターで学習者の質問や相談に対応するために、学習者の状況を把握して支援することが困難な場合がある。また、学習者がプログラミングでつまづいても挙手をしてメンターに相談しない場合がある。

そこで、メンターが演習中に学習者の様子を把握し

やすくするための学習状況把握支援システムやメンターの代替になり得る対話型のシステムなどが提案されてきた。学習状況把握支援システムには、エラーログや操作ログから学習者のつまづきを検知してメンターに表示するシステム [2] や、学習者の進捗状況を整理してメンターに表示するシステム [3] などがある。メンターの代替になるシステムにはシステムと学習者の中でチャットで会話を行うことによってつまづき解決を図るシステム [5] がある。

一方、近年プログラミング導入教育においてペアプログラミングの利用が増えている。McDowell ら [4] はプログラミング導入教育においてペアプログラミングは、落第可能性のあった者の単位取得、コードの質の向上に効果があったと述べている。

本研究では、ペアプログラミングを使った演習形式の講義を対象とした学習状況把握支援システムの提案を行う。

2 関連研究

2.1 学習状況把握支援システムに関する過去の研究

いくつか既存の学習状況把握支援システムに関する先行研究をあげる。

市村ら [2] の研究は、つまづいている学生の早期発

A Support for Understanding Learning Status when Using Pair Programming.

Megumi Unno, Kei Ito, 公立はこだて未来大学情報アーキテクチャ学科, Dept. of Media Architecture, Future University Hakodate.

見と多くの学生が共通に抱える問題の発見を目的として Web ブラウザ上で動く学習状況把握支援システムの提案を行っていた。市村らのシステムは、エラーログと操作ログをサーバ側で収集して解析し、メンターにわかりやすく提示するシステムである。市村らのシステムは、エラーが出た場合はつまづきとして検知できるが、計算結果が違うというような意図した出力でないという問題には対応していない。

井垣ら [3] の研究は、進捗が相対的に遅れている学習者を早期発見しメンターが支援に入ることを目的として教育支援システムを提案していた。このシステムでは、コード数、課題ごとのコーディング時間、単位時間当たりのエディタ操作数、課題ごとのエラー継続時間をもとにコーディング過程の可視化を行うことで、進捗が遅れている学生の支援を行った。井垣らのシステムは、課題 1 問がより複雑になり、難易度が上がった場合に進捗が管理しにくくなる可能性がある。

2.2 メンターの代替システムに関する先行研究

松井ら [5] の研究は、メンターが学習者に話す内容が知識の伝達と思い出しを促す質問に分類されることを示した。そして、チャットでの対話によって学習者へ知識の伝達や思い出しを促す質問を行い、つまづき解決を試みるシステムを開発した。松井らのシステムでは、学習者が特定の単語をシステムに伝えることによって対話が始まる。この場合、学習者のつまづきの原因が基礎知識の不足や忘却であるつまづき解決に限られてしまう。

2.3 ペアプログラミングにおけるつまづき分析に関する過去の研究

ペアプログラミングにおけるつまづき解決の成功と失敗にみられる会話の違いが平井ら [1] によって分析されている。それによると、つまづき解決に失敗するペアは、成功するペアと比べて会話の発話長が長く、説明の繰り返しが多い。この特徴から、演習形式の講義中につまづきを検知できる可能性があると考えられる。

3 本研究の目的とアプローチ

本研究の目的は、メンターが効率的に学習状況を把握し、学習者に支援が行うことができるようにすることである。これを実現するために、以下の 3 つのことは行うシステムを開発する。

- 学習者のつまづき検知
- つまづき解決の難易度ごとに分類・選別
- 学習状況を整理して表示

4 予備調査

学習状況把握支援にあたって、メンターが実際に対応している学習者からの相談について知るために予備調査を行った。以下は、実施した予備調査の質問項目である。

質問 1 受講生によく聞かれる質問はどのようなものですか

質問 2 受講生に聞かれた質問のなかで印象に残っているものはありますか

質問 3 TA として対応しなくても学習者自身で解決できそうな質問はどのような質問だと思いますか

質問 4 対応に時間がかかる質問はどのような質問ですか

質問 5 「対応に時間がかかる質問」はどのようなになる/することがあれば、現在よりも短時間で対応することができると思いますか

質問 1 は、質問される頻度の高い優先的に対処しなければならない可能性のある質問を知るための質問である。質問 2 は、学習者から TA にされる質問を知るための質問である。質問 3 は、学習者からの質問の中で TA が答えなくても良い可能性のある質問を知るための質問である。質問 4 は、対応に時間がかかる効率化が求められる質問を知るための質問である。質問 5 は、TA が考える対応に時間がかかる質問の原因を知るための質問である。

4.1 予備調査対象

著者らの所属大学で1, 2年時に開講される演習型のプログラミング授業でTA経験がある人にアンケート調査を行った。

4.2 予備調査結果

本稿執筆時点までに9件の回答が得られた。回答結果全体は表1, 表2, 表3に示す。主要な回答を以下で述べる。

対応しなくてもよいと思った質問として, () や {} の非対応などのケアレスミス, エラー内容を確認することによって解決できそうなもの, 基礎知識の不足によるものという回答が共通して見られた。回答に時間がかかる質問として, 学習者の基礎知識が不足しているときの質問, コードが複雑な場合の挙動に関する質問や, メンターがあまり遭遇しないエラーに関する質問が挙げられた。また, 回答に時間がかかる質問はどのようにすれば対応時間が短縮できるかという質問では, ケアレスミスをチェックするようリストを使った対応や, 学習者が自身の考えをメンターに効果的に伝えることをすれば対応時間が短縮できるという回答があった。

4.3 予備調査考察

調査結果から, エラー内容やケアレスミスなどの確認不足が原因の学習者自身で解決できそうな質問もメンターが受けていることが分かった。これらは学習者にエラー内容やケアレスミスの確認を促すことで, メンターが受ける質問を支援の必要性が高いものに絞ることができると考えた。

また, 今回のアンケートで学習者の基礎知識が不足している場合についての言及が多数見られた。基礎知識の不足は, TAに聞かなくても自己解決可能なことであり, 対応に時間がかかるという認識をTA経験者が持っていた。このことから, 基礎知識の不足に起因するつまづきをTAが対応しなくてもよいつまづきとして選別する必要があると考える。選別方法としては, 学習者にシステムを通して簡単な問答を行い, 学習者が基礎知識を思い出すことを促したり, 基礎知識が得られる媒体を紹介したりすることが考えられる。

そして, 回答時間を短縮するためには学習者が自身の考えをメンターに効果的に伝えられれば対応時間が短縮できるという回答から, メンターが学習者の思考の把握に多くの時間を費やしていることが分かった。ゆえに, メンターが学習者の思考を把握することを支援することが必要だと考えた。

5 提案システム

本研究では, ペアプログラミングを利用した演習形式の授業での学習状況把握支援システムの提案を行う。システムのおおまな構造は, 図2に示す。予備調査や先行研究から, 本研究で提案する学習状況把握支援システムは, 以下の3つの機能を実装することを検討している。

1. 学習者のつまづき検知機能

学習者のつまづきを検知する機能である。ここにおけるつまづきとは, 学習者が課題プログラム作成で行き詰ることである。この機能はエラーログ操作ログと, 発話の繰り返しからつまづきを検知することを検討している。

2. 学習者のつまづき選別機能

学習者が自己解決可能なつまづきと, メンターの支援が必要なつまづきを選別する機能である。学習者とシステムで対話を行うことによって, 学習者自身で解決が可能な問題か, メンターを呼んで解決する必要がある問題か推定する。そして, 学習者自身で解決が可能な問題ならば, 学習者がつまづきを解決するための簡易的なアドバイスを行う。メンターを呼ぶ必要があるつまづきならば, メンターに通知する。

3. 学習者の進捗状況表示機能

この機能は, メンターが進捗の遅れている学習者を把握するために, 一定時間ごとにエラーログや操作ログなどの情報を収集し, 常にメンターが学習者の進捗状況を見ることができるようになる。また, 全体の進捗を把握することにより, 受講している学習者全体が共通してつまづいている個所が発見できる可能性がある。

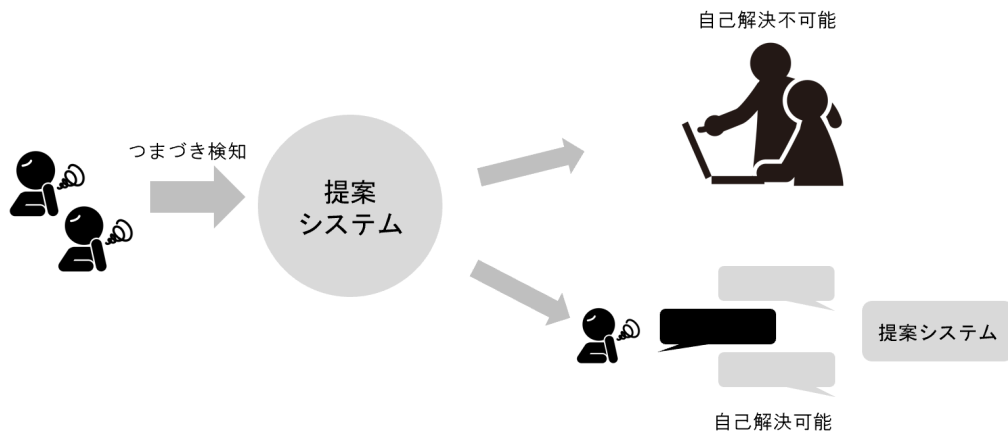


図 1 提案システムをつまづき検知から選別するまでの図

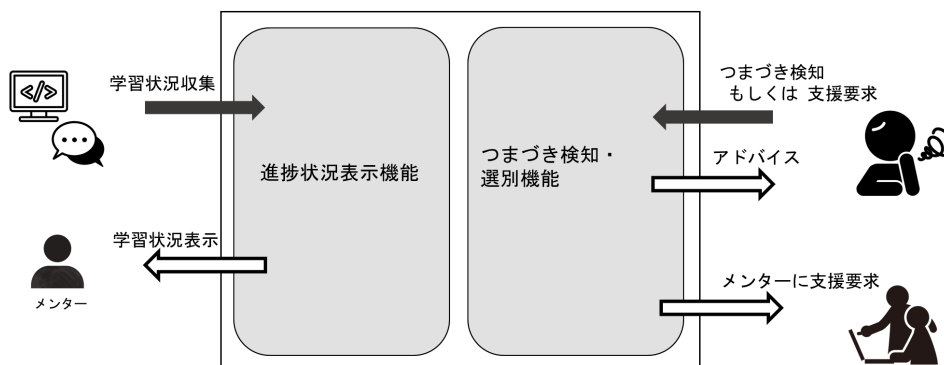


図 2 提案システムの概要

5.1 提案システムの理想シナリオ

提案システムをつまづき検知機能とつまづき選別機能が理想的に働いた場合、学習者につまづきが発生するとシステムで把握し、学習者自身で解決できそうなつまづきだった場合、システムが学習者のつまづき解決を助ける簡単なアドバイスをしてつまづき解決をする予定である。選別機能が理想的に働いている様子を図1に示す。このとき、メンターに対して、学習者がつまづいていることを表示するが、学習者の支援をするように誘導はしない。一方、検知したつまづきが

学習者自身での解決が困難だと判断した場合、システムはメンターに学習者の支援をするように誘導する。

6 今後の課題

つまづき検知をエラーログ、操作ログから取得していることを検討しているが、どのような条件でつまづきとするか決める必要がある。現在、同様のエラーが一定時間表示されていた場合、操作が一定時間されなかった場合をエラーと定義することを予定している。同一のエラーが表示される継続時間や、操作が止まっ

ている継続時間は、仮で時間を設定してシステムを実装したのち、評価実験をすることで決めていく予定である。

また、ペア間の説明の繰り返しからつまずきを検知する予定である。実現方法として、音声の文字起こしをして、説明の繰り返しが複数回発生した場合つまずきとすることを検討している。この検知機能は、説明の繰り返しが判断するシステムが実現後につまずき検知のタイミングの妥当性を評価する評価実験を行う予定である。

つまずき検知方法の確立の後、システムのプラットフォームを選定し、開発を進めてメンターの学習状況把握の効率化を図る。

7 まとめ

本研究では、少ないメンターで演習形式のプログラミング講義でメンターが学習支援を効率よく行うためには、学習状況把握支援が必要であると考え、学習

状況把握支援システムの提案を行った。今後は、つまずきの定義の妥当性の検証や、システムの開発を行っている。

参考文献

- [1] 平井佑樹, 井上智雄: ペアプログラミング学習における状態の推定 つまずきの解決の成功と失敗に見られる会話の違い, 情報処理学会論文誌, Vol. 53, No. 1(2012), pp. 72-80.
- [2] 市村哲, 梶並知記, 平野洋行: プログラミング演習授業における学習状況把握支援の試み, 情報処理学会論文誌, Vol. 54, No. 12(2013), pp. 2518-2527.
- [3] 井垣宏, 斎藤俊, 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(2013), pp. 330-339.
- [4] MacDowell, C., Werner, L., Bullock, H., and Fernald, J.: The Effects of Pair-Programming on Performance in an Introductory Programming Course, *ACM SIGCSE BULLETIN*, Vol. 34(2002), pp. 38-42.
- [5] 松井大成, 伊藤恵: プログラミング行動中の学習者と指導者による対話に基づいた初学者支援システムの提案, 公立はこだて未来大学卒業論文, (2019).

表 1 予備調査回答一覧 (1)

質問番号	質問内容
質問 1	<ul style="list-style-type: none"> ・ Eclipse が何かしらの原因で開けない、メニューが見えない ・ 実行結果はあっているが、提出したら間違っているとされた (Export 時に間違っている、何かしらのケースを想定していない) ・ フィールドやコンストラクタなどの使い方・概念の確認 ・ここまではできたが、ここから先何をやればいいのか分からない ・ 関数や構文について、こういう使い方であると理解したがそれが合っているかコードを書いたがチェックプログラムを通らない <p>指示はわかるがこれをどのようにコードに落とし込めばいいのか分からない</p> <p>arduino にスイッチを接続した際にうまく動かない、arduino のコンパイルエラーが出る。配線があるか確認してほしい。</p> <p>一番聞かれたと思う質問は「自分の書いたプログラムが思うように動かないのだが、どこに原因があるか？」という内容の質問でした。</p> <p>他には「C のポインタやアドレスって何ですか？」や「何でこのエラーが出ているか分かりますか？」という質問がよく聞かれたと思います。</p> <ul style="list-style-type: none"> ・ 演習の内容をどのように実装していいか分からない ・ eclipse のパッケージなどを見ることができている部分が消えてしまった ・ HOPE に提出した際にエラーメッセージが表示されたが、修正箇所が分からない <p>「プログラムがエラーを起こしてうまく動かない」という内容</p> <p>この問題の解き方が分かりません。</p> <p>この問題はここまではわかったんですが、ここから先はわかりません。</p> <p>この問題の回答の仕方はこれで合っていますか？</p> <p>何をしていいかわからない</p> <p>list に要素を追加する方法がわからない (配列との違いがわからない)</p> <ul style="list-style-type: none"> ・ 問題の意味がわからない ・ 配列とは何か ・ 作業用変数とはなにか ・ コンストラクタとはなにか ・ インスタンスとはなにか ・ 変数の初期化とはなにか ・ 関数 (メソッド) の作り方 ・ 引数とはなにか ・ 戻り値とはなにか

表 2 予備調査回答一覧 (2)

<p>質問 2</p>	<p>ここまでではできたが、ここから先何をやればいいのか分からない プログラミング基礎でポインタを使えると何が嬉しいのかという質問 すみません、あまり思い出せませんでした。</p> <p>あまり聞かれることはなかった質問ですが、既に問題が解けた状態で「自分の書いたプログラムは正しいというか、変なところはないか?」や「これで提出しても大丈夫なプログラムなのか?」、「他にどのような書き方があるのか?」というような質問がありました。</p> <ul style="list-style-type: none"> ・全空港の数と全空港の総滑走路数と平均の滑走路数を表示する課題にて、平均の滑走路数を求めることができているのに、全空港の数が表示できないという質問→平均の滑走路数をどのように求めたかを確認したところ解決した ・コンストラクタの作成でエラーが出たという質問→コンストラクタ内で不要な処理をしていたことが原因だったので HOPE の課題ページを確認するように促したところ解決した <p>配布されたひな形のコードを参考にしてうまくいかないという内容の質問で、原因がひな形のコードのバグであったこと 特になし</p> <p>Java の this が何を指すのかわからない スタックオーバーフローってエラーが出たけどなんのことかわからない クラスを二つ選択してエクスポートするやり方がわからない</p> <ul style="list-style-type: none"> ・何をすれば良いかわからない
<p>質問 3</p>	<p>資料のコードを写すだけで動くコードのミス、大体の場合が打ち間違っって正常に動作しない。 授業中に説明されている内容に関する質問は学習者自身で解決できそうな質問だと思っていました。 なので、「int 型や double 型って何ですか?」や「for 文はどういう動きをするんですか?」というような質問が来たときは、自分で簡単な説明をした上で、講義で使用しているスライドに詳細な説明があるので、そちらを参考にすると良いと伝えていました。</p> <p>eclipse や HOPE の特定のエラーはエラー内容から修正箇所を知ることができるので学習者自身で解決ができると思います。</p> <p>また、課題ページを見返すことで解決が可能な質問も学習者が行き詰まった段階で見直すようにすることで解決が可能だと思います。</p> <p>エラーメッセージを学生が読んでいないと思われる質問 特になし</p> <p>変数名のスペルミスは、なぜかエラーが起きる、コンパイルが通らない時のチェック項目などに入れておけば自己解決可能だと思う。</p> <p>講義資料や問題文に書いてあること全般。</p> <p>また、既習の範囲で、プログラミングに関する基礎的な知識について</p>

表 3 予備調査回答一覧 (3)

<p>質問 4</p>	<p>何をやればいいのか分からない (ヒントを出しても出したヒントを理解できない程度に基礎知識がない場合)</p> <p>うまくいかない原因が複数あるような質問 (バグ、構文ミス、そもそも講義内容を理解していないといった要因が複数絡む様なもの)・PC のトラブル関連の質問・学習者が解いている問題外の要因に起因する質問</p> <p>見たことのないエラー文. おそらくコードではなくハードウェア関係の問題. そもそも arduino の機器自体に問題がある場合が多いのでどうしようもない.</p> <p>複雑なプログラムを記述しているときの「予想している挙動と違うのはなぜか?」という質問や、自分でもあまり見たことのないエラーで詰まっているときにどうやって解決するかという内容の質問が来たときは対応に時間がかかっていました.</p> <p>試行錯誤した結果, 学習者自身も内容が分からなくなってしまった際にされる「 の課題の が分かりません」のような質問</p> <p>学生にアイデアがない質問「～が全く分からない」など</p> <p>解答のきっかけが全くつかめないのですが, この問題はどう解いたらいいですか?</p> <p>そもそも前提となる基礎知識が欠けている場合、「なんだかよくわからない」という曖昧な質問がくるので, 前提知識から教える必要があるため時間がかかります。例えば, Java のクラス概念があまり理解できていない場合, 質問の対応の前に, そこを 1 から教えて理解してもらう必要があります。</p> <ul style="list-style-type: none"> 何をすればよいか分からない
<p>質問 5</p>	<ul style="list-style-type: none"> 学習者が, プログラミングを行う上で基礎的な概念を理解する 教授者が, 口頭だけでなく図的な解説をする <p>初めの一つに関してはわからないところのヒアリングをしっかりすれば, 時間をある程度短縮できる気がします。しかし, 講義内容や今までにやってきたことを理解していない学生に関しては手の打ちようがないです。PC やその他のトラブルに関しては事前に一律の環境でコードを実行できる様にすると問題が起りにくくなり, 対応時間も短縮できると思います。</p> <p>一度見たことのあるものであれば, 記憶の海を遡って対応できるかもしれない。確証はないです。</p> <p>初歩的な部分でミスをしていることもあるので, 「; を忘れていないか」や「() やの数があるか」、 「全角の文字やスペースが混ざっていないか」などの質問する前に自分で確認することのリストみたいなものがあれば, 無駄に時間をかけてしまうことが減ると思います。</p> <p>また, コンソールに最終的に出力するものだけを出力するのではなく, 処理の途中でどの変数にどんな値が入っているのかを printf() とかを使って確認するようにすれば, 受講生/TA ともに原因の特定とその解決にかかる時間を減らせるのではないかと思います。</p> <p>学習者がどのように考えたかを TA に上手く説明することができれば原因が分かるまでの時間が短くなると思います。</p> <p>全体的な方針や考え方のみを伝えて, 具体的な実装方法は学生に考えさせる</p> <p>解答におけるきっかけをヒントとして公開してくれればある程度はましになるかなと思います。</p> <p>その演習の回で理解すべき部分が曖昧の状態では, 次の演習回に進めないようにする。</p> <p>もしくは, 基本的な問題文をよく読むことを促す</p>