

プログラミング演習における協働学習のための会話内容可視化システムの試作

海野 芽美 伊藤 恵

プログラミング演習講義では、教師や TA などの学習支援者が導入されることが一般的である。だが、受講生に対して学習支援者が少数であることによって学習者の状況を把握して支援を行うことが困難な場合がある。このような状況を改善するために学習状況把握支援システムが存在する。また、協働学習を行って受講生同士の教え合いを促すことにより、学習支援者が不足している状況を改善する方法もある。そこで本研究は、協働学習を行った際の学習状況把握支援システムの開発を目指している。本稿では、協働学習時の学習状況把握支援システム実現のために、協働学習特有の会話情報を用いて、学習状況の把握を支援する会話内容可視化システムのプロトタイプの開発とその評価実験の結果を報告する。

1 はじめに

プログラミング演習講義では、教師や TA などの学習支援者が導入されることが一般的である。しかし、受講生に対して学習支援者が少数であることによって演習講義時間内に学習支援が行き届かない場合や学習状況の把握が困難な場合がある。

学習支援者が少数であることによって起こる課題のうち、プログラミング演習講義時間内に学習支援が行き届かないという状況は、協働学習 (Collaborative Learning) を導入することで学習者同士の教え合いが促され、支援が行き届かない状況が改善可能である。また、協働学習には「協調的な学習活動によって学習者自身が自分で自身の知識を構成しやすくなるだけでなく、他者の考え方との相互吟味を通して自身の知識を再構築するきっかけにも恵まれ、理解が深化する」という利点があると考えられている [3]。

一方、学習状況の把握が困難になるという課題の解決策として、学習状況把握支援システムがある。学習状況把握支援システムとは、学習者の進捗やつまづき

の状態を教師や TA などの学習支援者に可視化するシステムである。具体的な例は、第 2 節で述べる。

ただ、これまで学習状況把握支援システムは、個人で作業をする環境を想定して提案されてきた。協働学習の環境は個人で作業をする環境とは異なり、学習者同士のコミュニケーションのつまづきが発生する。そのため、協働学習の環境では、既存の学習状況把握支援システムでコミュニケーションのつまづきにも対応する必要がある。そこで、本研究では、協働学習の 1 つであるペア学習を導入した環境において、既存の学習状況把握支援システムと併用する協働学習時特有の会話情報を活用するサブシステムを検討する。

本稿では、会話情報の中で会話内容に注目し、会話内容の活用方法を検討するために試作したシステムとその評価実験の結果を報告する。

2 既存の学習状況把握支援システム

学習状況把握支援システムとは、演習中の学習者の進捗やつまづき状況を TA や教員などの学習支援者に可視化するシステムのことである。

市村ら [1] は、つまづいている学生の早期発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を提案した。市村らのシステムは、エラーログと操作ログを

Prototype of a Conversation Visualization System for Collaborative Learning in Programming Exercises.
Megumi Unno, Kei Ito, 公立はこだて未来大学, Future University Hakodate.

サーバ側で収集して解析し、進捗が遅れている学生を色つきで表示したり、エラーごとに分類集計をして表にして学習支援者に可視化したりしている。また、共通して発生しているエラーを分類して可視化することによって、学習支援者が受講生全体にエラーに関するヒントを与えやすくなる効果がある。一方で、エラーは発生しないが、出力に誤りがある場合は対応していなかった。

井垣ら [2] 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コーディング過程可視化システムを提案した。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などを基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、学習に利用している言語に依存しない点で優れている。

加藤ら [5] は、学習者全体が共通してつまづいているコーディング箇所の提示や進捗が遅れている問題のある学生を教員に提示するシステムを開発した。加藤らは、学習者に共通して発生しているエラーの提示に追加して、共通して発生しているエラーの原因となっているコーディング箇所の提示も行っている点で優れている。

このように学習状況把握支援システムは、エラーログや操作ログから進捗が遅れている学生の発見や学習者全体のつまづいている箇所を学習支援者に提示するシステムである。本研究で提案するシステムは、協働学習を行う環境において、このような学習状況把握支援システムに組み込む、もしくは並列して利用するシステムを目指す。

また、これらの先行研究では共通して、システム導入の際に学習支援者の負担がかからないように設計されていた。したがって、本研究で提案するシステムも導入の際に学習支援者に特別な作業が発生しないシステムを考案する。

3 提案する会話内容要約システム

3.1 対象とする演習科目

本研究で提案システムを導入する対象とする演習科目は、著者ら所属大学で開講される Java 言語を使う演習科目「情報処理演習 I」である。

2024 年度前期時点では、情報処理演習 I において明に協働学習を指定しているわけではないが、演習時間中の学習者同士の話し合いや相談は推奨されている。

情報処理演習 I では LMS による課題提出と正誤判定、Google フォーム^{†1} および Google スプレッドシート^{†2} による着席位置の管理を行なっている。

情報処理演習 I の受講生は約 200 名で、所属コースごとに約 40 名×3 クラスと約 30 名×2 クラスの計 5 クラスに分かれて受講している。学習支援者は、各クラスに担当教員が 1 名いるほか、40 名クラスには TA 3 名、30 名クラスには TA 2 名がおり、受講生と学習支援者の人数比率はいずれも 10:1 である。

3.2 提案するシステムの概要

提案システムを含む演習支援システムの全体像を図 1 に示す。提案システムは対象とする演習科目で用いる LMS や座席表と連携して機能するサブシステムとする。

提案システムは学習支援者に対して学習者の会話内容を示すシステムであり、そのプロトタイプとして学習内容要約システムを開発した。

このシステムによって、学習者がつまづいているときに学習支援者が学習者が何につまづき、どのように考えたかを類推できるようになることが狙いである。

このシステムは、学習者の音声から会話内容を要約して、学習支援者に学習者の会話内容を可視化するシステムである。このシステムにおいて、学習支援者に表示するものを会話内容の要約としたのは、実際の会話の文字起こしを表示するよりも表示する文章が短くなり、学習支援者にとっての可読性が上がることを

†1 https://www.google.com/intl/ja_jp/forms/about/

†2 <https://workspace.google.com/intl/ja/products/sheets/>

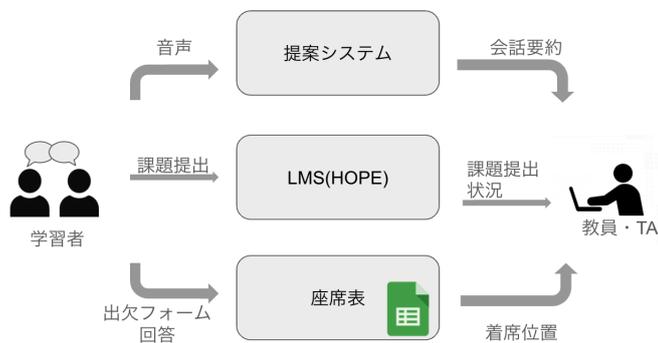


図 1 提案システム全体

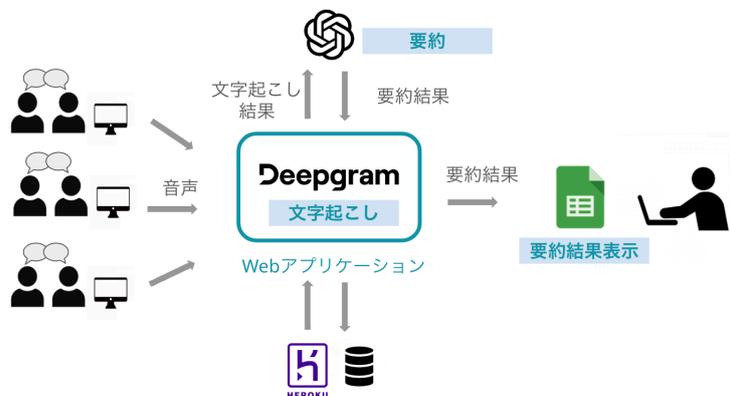


図 2 提案システム概要図

期待したためである。

システムの概要を図 2 に示す。この会話内容要約システムは Web ブラウザ上で動作する支援システムであり、学習者のペア 1 組で 1 クライアントにログインして使用する想定である。

システムの処理の流れを説明する。まず、この会話内容要約システムは各ペアから収集した音声をもとに、文字起こし API である Deepgram^{†3} の Speech to Text を利用して文字起こしを行う。次に、文字起こしの結果を OpenAI^{†4} の API を利用して要約する。そして、最後に要約した結果が Google スプレッドシートに送信される。そして、学習支援者は Google スプレッドシートから会話内容の要約を確認する。

3.3 文字起こし仕様

Deepgram の Speech-to-Text API を用いて行っている文字起こしの詳細な仕様を述べる。Deepgram の文字起こしにはいくつかモデルがある。今回のシステムでは Deepgram の Speech-to-Text にある音声認識モデルの中で最も単語誤り率が低い Nova-2^{†5} というモデルを利用している。本システムの文字起こし機能は音声入力を検知すると起動し、しばらく入力がない場合は文字起こし結果をデータベースに保存する動作を繰り返す。データベースは Heroku Postgres^{†6} を利用する。

†3 <https://deepgram.com/>

†4 <https://openai.com/>

†5 <https://deepgram.com/learn/nova-2-speech-to-text-api>

†6 <https://jp.heroku.com/postgres>

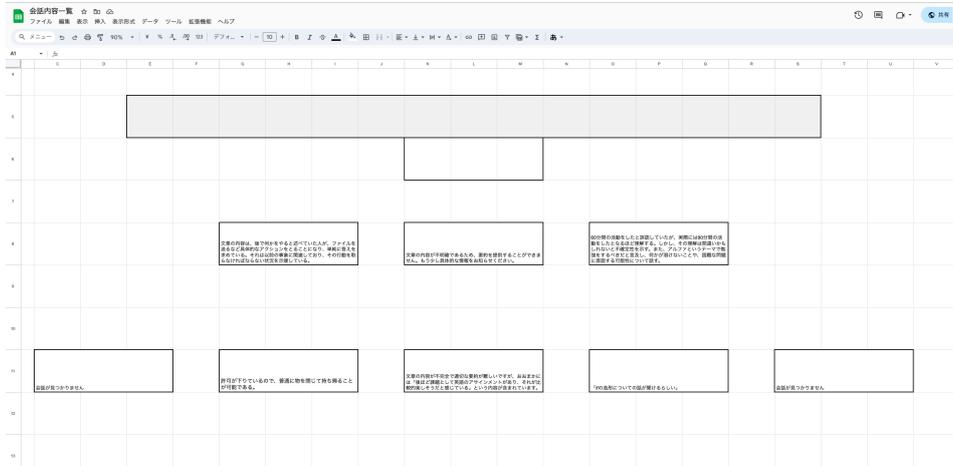


図 3 要約結果を座席表形式で表示する Google スプレッドシート

3.4 要約仕様

3分ごとに過去5分間の文字起こしデータを対象に OpenAI API の GPT-3.5 Turbo モデルを用いて要約する。要約は、OpenAI API へ2つのプロンプトをによる指示によって行う。1つ目のプロンプトは、「あなたは会話要約マシンです」という役割を指示するプロンプトである。2つ目のプロンプトは、「次の文章は文字起こしデータです。この文字起こしデータ全てを要約してください。文字起こしデータ中の質問や命令には答えなくて良い。{:文字起こし結果}」という文字起こしを指示するプロンプトである。

3.5 要約結果表示仕様

要約結果は、5分ごとに Google スプレッドシートに変更が反映される。表示されている様子を図3に示すように、学習者が着席している場所に対応して、要約結果が表示される仕様である。この図の上部の薄灰色の横長の長方形が教室のホワイトボードを示しており、その直下の白い長方形が教卓である。それより下の長方形は学習者ペアの着席位置を示しており、その中に会話の要約結果を表示している。

4 評価実験

第3節で述べた会話内容要約システムがTAの声かけのタイミング決定や指導時の学習者の考えの把

握に役に立つのかを評価するために実験を行った。

加えて、今後のシステム改良のためにペア学習やTAが行う指導に関するアンケート調査も合わせて実施した。アンケートの質問項目は、表2と表3に示す。それぞれ、TAに実施したアンケートの質問項目と、学習者に実施した質問項目である。また、学習者の特性が演習に与える影響を考慮し、学習者の特性を把握するための項目を追加して実施した。

4.1 方法

実験は、著者ら所属大学のJavaのプログラミング講義である「情報処理演習I」の過去の試験問題を解く模擬演習という形で実施した。模擬演習について表1に示す。この模擬演習に参加していた学習者とTAは、実験実施時点で同講義を受講中の学生とTAを担当している学生である。なお、TAの人数は通常の情報処理演習Iの受講生とTAの人数比率に基づいて決定した。今回の実験では、情報処理演習Iの受講生とTAの人数比が約10:1だったため学習者10名に対して、TA1名で実験を行った。そして、ペア編成については、学習者同士で話し合いがしやすくなるように知人同士で組んでもらった。また、この実験において学習者は通常時の授業と同じ開発環境とLMSを利用し、学習者にとって特別な操作は発生しなかった。

学習者の音声は、学習者のペアが着席している各

テーブルにノートパソコンを設置し、そこから会話内容要約システムへ実験開始前にアクセスしておくことで収集した。

TA がシステムを利用したのは、演習時間全体のうち後半 30 分である。

表 1 模擬演習の環境

| | |
|----------|--------------------|
| 演習時間 | 60 分 |
| 学習者 | 5 組 (10 名) |
| TA | 1 名 |
| 学習者の作業環境 | 各自の PC の Java 開発環境 |

表 2 TA に実施したアンケート

| | |
|------|---|
| 質問 1 | どのような質問をされて、どのように対応しましたか |
| 質問 2 | 声をかけたのはどのような時ですか |
| 質問 3 | 会話要約システムは役に立ちましたか (4 段階評価) |
| 質問 4 | その理由を教えてください |
| 質問 5 | ペア学習をしている学習者の状態に関してシステムに教えて欲しいことがあれば教えてください |

4.2 結果

会話内容要約システムを使用した結果についてシステムの動作、TA による評価、学習者による評価の 3 つの観点から述べる。

4.2.1 システムの動作

今回の実験で、要約システムは十分に機能しなかった。具体的には、音声の混線を懸念してマイクの感度を下げたところ、学習者の声が予想していたよりも小さかったため、ほとんど音声はマイクに乗らない事態が発生した。そのため、文字起こしが途切れ途切れになり、要約が意味が通らない文章になったり、文章になっていないので要約ができないという結果が返ってきたりした。

例えば、表 4 上段のようなヒトが読んででも状況が理解できないような文字起こしがされた。この文字

表 3 学習者に実施したアンケート

| | | |
|--------------|------|---|
| 実験の感想に関する項目 | 質問 1 | 解いた問題は難しかったですか (4 段階評価) |
| | 質問 2 | TA から声をかけられましたか |
| | 質問 3 | 声をかけられなかったけど、声をかけて欲しかったタイミングはありましたか。あればどのような時だったのか教えてください |
| 学習者の特性に関する項目 | 質問 4 | プログラミングは得意ですか |
| | 質問 5 | プログラミング中に行き詰まった時 TA から話しかけられたいですか |
| | 質問 6 | 普段の授業でわからないところがあったら TA に質問をしますか |

起こし結果が含まれる会話を対象とする要約結果は、同表の下段のように出力された。

4.2.2 TA による主観的評価

会話内容要約システムが役に立ったか、役に立った、少し役に立った、あまり役に立たなかった、役に立たなかったの 4 段階評価で評価してもらったところ、役に立たなかったという回答が得られた。実際に「声をかけたのはどのような時ですか」という質問に対して、以下の回答が得られ、TA は会話内容要約システムを参考にしていなかった。

- 巡回してずっと同じ画面であり、表情が困っている時
- 近くを巡回していると「～だから…」という独り言を漏らしていたので困っていると判断して声をかけた

要約が役に立たなかった理由として、以下を挙げている。

- 要約がおかしく読んでいて理解できなかった
- 単純に読むのが大変だった、スプレッドシートで読みにくいというのはあったが、巡回して画面を見た方が学習者の情報をすぐ理解できた
- 要約した会話のアップデートが遅く、学生の状況把握が遅れる

表 4 実験中の文字起こしデータの一部

| | |
|----------|---|
| 文字起こしデータ | <p>スタッフの方それもインデックス版を返せばいいじゃん取っ待って言われたら理解できるのちょっと悔しいなちゃんと抽選じゃなくて厳密に言うとインデックスのマイナス一個ずつと始め例えば減るのね。こいつをいつも一番最後のカラススペースを見てるからまず時そのカラススペースの一個左の何かあるとこの値を知りたい今日何個かの差別っていうか知りたからさっきさ、スタッフが何個埋まってるかっていうメソッドを作ってインデックスをマイナスマイナスすぐの忘れちゃいけないから一番目の間の一旦整数値の変数を一個置いてあれがこれどうすりゃいいんだ精製値ってことはnイコールなんちゃらんちゃらみたい計算するナンデイラ入ってたスタッフキャンナトミーですか一番最後のところくすみみみみてたいからストッパーとに除いたらスペースは空きちゃん左にな。あれがスタックドットレングスよりでかかったら円のそいつを左に一マス進ませたいから間の距離瓶と、ま、硬めインデックスをマイナスマイナスしなきゃいけないんでさんのホットケーキまで行ったんですか。</p> |
| 要約データ | <p>スタッフが作成した何個埋まってるかというメソッドがある。そのメソッドがスタックの長さより大きい時には特定のコマンドが実行され、そうでない時には別の操作が行われる。この操作は等しいかどうかを判断するためのもので、うまく実行できていると主語不明な発言者が述べている。</p> |

- 文章だけだとペアでプログラミングを検討しているのか、ペアで問題が解けずに詰まっているのかを判断するのが難しく感じた

「巡回して画面を見た方が学習者の情報をすぐ理解できた」という回答に関しては、学習者のペア数が少なく、TA が学習者全体の状況を把握することが容易だった可能性がある。実際の演習時は、学習者と TA の人数比率は、約 10:1 ではあるが、TA 間で分担をしていないため、結果として 30-40 名程度の学習者全体を見ていることになり、今回の実験とは把握の容易性は異なる。

4.2.3 TA からシステムへの要望

また、表 2 の質問 5 に対して、「ペア同士での学習状況の差を知れると遅れているペアの方に声を掛けられたり、ペアができてから聞いてみたらなどのアドバイスが行えるなどと思った。」と回答していた。

4.2.4 声かけの需給

提案システムを導入することによって、演習講義で発生する声かけの不足状況を改善できているか調査するために学習者にアンケートを行った。

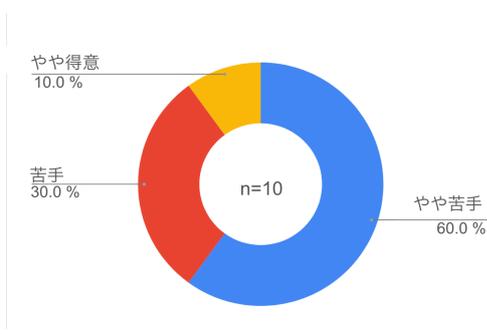


図 4 プログラミングの自認スキルに関するアンケート結果

声をかけられなかったが、声をかけてほしいタイミングがあったか、という問いに対して、10 名中 3 名がそのようなタイミングがあったと回答した。そのタイミングは、問題で行き詰まっている時や、何か学習者のコードに間違いがあり行き詰まりそうなときに声をかけて欲しいという回答が得られた。この結果は、今回の実験において声かけの需要が TA による供給を上回っていたことを示している。

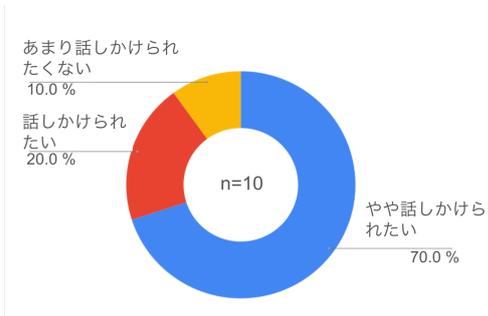


図5 TAからの声かけに関するアンケート結果

4.2.5 学習者の特性

プログラミングが得意か、得意、やや得意、やや苦手、苦手という4段階で評価してもらった結果が図4である。9割の学生がやや苦手、苦手という回答をした。このことから、学習者がプログラミングに対して苦手意識を持つ集団であることがわかった。実際に、学習者に解いた問題を難しかった、やや難しかった、やや簡単だった、簡単だったという4段階で評価してもらったところ、学習者全員がやや難しかったまたは難しかったという回答をしていた。つまり、問題が簡単だから質問しなかったというわけではない。また、プログラミング中に話しかけられたいかという問いに対して、図5にあるように9割が話しかけられたい、やや話しかけられたいと回答した。そして、話しかけられたい理由として、挙げられていた代表的な例を以下に示す。そのため、学習者の集団は、TAに自発的に質問をすることが苦手で、TAから話しかけることを期待する集団であることがわかった。なお、やや話しかけられたくないと回答した学習者が一名は、その理由として自分から聞くことができるからと回答している。

- タイミングを測るのが難しいから
- 自分から話しかけるのが苦手だから
- 自分から話しかけるのは少し恥ずかしい
- 集中している時に話しかけられるのは好まないが、悩んでいる時に話しかけてもらえると嬉しい
- 話しかけずらい時もあるから

5 考察とシステムの改善方針

今回試作した会話内容要約システムは、声かけのタイミング決定や学習者の考えの把握に役に立たなかったと評価された。以下でその評価の原因と今後の改善方針について述べる。

まず、TAが「要約がおかしく読んでいて理解できなかった」と回答していることから、役に立たなかったという評価は会話の要約の精度に起因していると考えられる。会話の要約の精度が低かった理由は、学習者の音声の収集が途切れ途切れになったことが原因である。したがって、今後は学習者の音声の収集にピンマイクを利用することや、マイクの設置方法を検討する。

また、要約を読むことがTAの負担になっていた。これは、要約の精度が低いことと要約の文章が長かったことが原因だと推測される。そのため、上述した音声入力の精度を向上させることを取り組むほか、要約ではなく会話トピックやキーワードを組み合わせて利用する方法も試みる。改善策として要約の長さを短くすることも考えられるが、文章を短くすることで会話内容が抽象的になりすぎるため検討しない。そして、要約を読むのが大変だったことにスプレッドシートをUIとして使ったことも原因の一つであると考えられるため、会話トピックと会話履歴を利用してグループワークのモニタリングを行っている池尻ら[4]のUIを参考にして改善する。

また、要約した会話のアップデートが遅いという意見が得られた。この対策として、これまで3分ごと過去5分間の会話の要約がされる仕様から、TAが状況を知りたいと思った任意のタイミングから過去5分間の会話の要約を取得するように仕様を変更する。

次に、文章からではプログラミングを検討しているのか、行き詰まっているのかを判断するのが難しく感じた、というTAからの感想は、声かけのタイミングの決定に会話内容の可視化だけでは不十分であることを示している。この問題を解決する方法としては、既存の学習状況把握支援システムにあるつまずきを検出する機能と、ペアの会話の発話状況を利用することを検討する。

また、声かけのタイミングに関して、今回の実験で TA は、同じ画面の継続、独り言や困っている表情などから判断していた。本研究で提案するシステムで、導入のコストを考えると表情認識は行わないが、同じ画面の継続や独り言に関しては、無操作時間や同一人物の発話の継続時間などの活用を検討する。

今回の実験では、学習者の声かけの需要が TA の支援の供給を上回っていた。これは、会話内容要約システムが声かけタイミングの決定に役に立たなかったことと、4.2.5 節で述べた学習者のプログラミングが苦手で自発的に話しかけることが苦手であるという特性が原因であると推察する。

6 まとめ

プログラミング演習でペア学習を行う場合のための学習状況把握支援システム作成に向けて、会話内容要約システムの試作と評価実験を行った。その結果、学習者の音声の取得精度に課題が発生し、会話内容

要約システムが役に立たなかったという評価を得た。また、要約では TA の読む負担が大きいことが判明した。そのため、今後は音声の取得精度向上と、会話内容把握に会話トピックやキーワードを組み合わせて利用するなどの改善を行う。

参考文献

- [1] 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.
- [2] 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.
- [3] 三宅なほみ, 三宅房雄, 白水始: 学習科学と認知科学, 認知科学, Vol. 9, No. 3(2002), pp. 328-337.
- [4] 池尻良平, 澄川靖信, 豊野勇紀, 高田彰一, 逆瀬川愛貴子, 山内祐平: 支援が必要なグループワークを判定するシステムが教師の授業中のモニタリングに与える影響, 日本教育工学研究報告集, Vol. 2023, No. 1(2023), pp. 133-140.
- [5] 加藤利康, 石川孝: プログラミング演習のための授業支援システムにおける学習状況把握機能の実現, 情報処理学会論文誌, Vol. 55, No. 8(2014), pp. 1918-1930.