

askTA : 消極性を考慮した オンライン演習講義支援システム

又吉 康綱 中村 聡史

大学で開講される初年次必修プログラミング教育では、TA が必要不可欠である。対面講義でもあった積極性を発揮できず質問できない受講生の問題は、COVID-19 の影響により大学の講義がオンラインになったことで、より大きな問題となっているといえる。また、それに加え質問の順番待ちや質問対応などの制御や、TA にとって質問に対応できるかどうかという精神的な負荷も大きな問題となりうる。そこで本研究では、受講生の質問へのハードルを下げつつ、TA の精神的負荷も軽減するため、受講生が TA に直接質問をするのではなく、受講生はシステムに対して質問を行い、また TA は質問を事前に確認し、対応可能な場合に呼び出して入室を促す手法を提案し、実装した。また、実際のオンライン講義で計 1600 分運用し、受講生および TA から高い評価を得ることができた。

TAs are indispensable for first-year programming education offered at universities. The problem of students who cannot ask questions due to their inability to be proactive has become a major problem in online lectures due to the impact of COVID-19. In addition, the control of waiting for questions and responding to questions, as well as the mental stress of whether the TA will be able to respond to questions, can also be major problems. In this study, we propose and implement a method in which students do not ask questions directly to the TAs, but ask questions to the system, and the TAs check the questions in advance and call the TAs when they are available. Our method can lower the barrier for students to ask questions, and also reduce the mental load of TAs. We have used our method for a total of 1600 minutes in an actual online lecture and obtained high evaluations from students and TAs.

1 はじめに

大学の初年次必修プログラミング教育では、数十人から数百人の受講生を一度に指導する必要がある。例えば、我々が所属している明治大学総合数理学部先端メディアサイエンス学科では、100 人以上の受講生がプログラミングの必修講義を同時に受講している。プログラミングは、受講生によって得意不得意や理解度などに大きなばらつきがあるうえ、基礎になるタイピング速度も違うため、多面的なサポートが必要となる。

我々はタイピング速度や基礎力向上に向けて、写

経とタイピングゲーム、動的な実行を組み合わせた typing.run[19] を実現し、実際に予習として課すことでその有用性を明らかにしてきた。一方、講義中のサポートを適切に行うには、TA(ティーチングアシスタント)の存在が不可欠である。一般的に、初年次教育においては TA の数は不十分であり、1 人の TA が受講生 10 人以上を担当することは珍しくなく、工夫が必要となる。

ここで対面型の演習講義では、質問したい受講生が手を挙げて TA を呼び、TA は受講生の画面と一緒に見ながら質問を解決する。受講生からの質問が多い場合には、手が上がった順番を考慮しながら対応していくことが可能である。しかし、質問の順番が TA の記憶だよりとなり順番を飛ばされることや、手を上げ続けることが大変などの問題もある。また、TA に対してこんなことを質問していいのかといった不安や、質問している様子を他者に見られることが恥ずかし

askTA : Online Practice Classroom Support System
Considering Passivity.

Yasutsuna Matayoshi, Satoshi Nakamura, 明治大学,
Meiji University.

コンピュータソフトウェア, Vol.39, No.1 (2022), pp.55-71.
[研究論文] 2021 年 2 月 27 日受付.

いなど、積極性を出せないことによる問題もあった。さらに、TA も質問で呼ばれ対応したものの、解決できず困るという問題もあった。

世界を激変させた COVID-19 の影響により、大学の講義がオンラインで行われることになったことで、講義において TA がいかに支援するかという点はより大きな問題となっている。一般的に、TA と受講生は多対多の関係で、受講生の数が圧倒的に多いため、オンラインコミュニケーション用のシステムにある挙手機能などを使って支援することは容易ではない。また、オンライン上にビデオ通話アプリケーションなどで質問部屋を用意して TA に待機してもらい、その部屋に受講生が入室することで質問対応する方法も考えられるが、質問部屋の中に誰がいるのか、TA がどういう状況にあるのか、どんなことを喋っているのかなどを把握することができず、入室をためらってしまうという問題もある。さらに、オンライン上では距離に違いがないため、順番待ちを行うのは容易ではない。一方、TA も得手不得手があるため、質問を即座に解決することができなくて時間を浪費することに加え、他の TA との情報共有が難しく精神的な負荷がかかるといった問題がある。さらに、質問の回答時間が伸びてしまうことで対応する効率が落ちてしまい質問待ちの受講生を多く待たせてしまうことに繋がる。表 1 は対面講義とオンライン講義の問題点を TA と受講生、共通の課題を整理したものである。

そこで本研究では、受講生が TA に行う質問のハードルが上がることと TA が精神的なプレッシャーを感じてしまうこと、質問順番の管理が困難なことに着目して、受講生の質問へのハードルを下げつつ、TA の精神的負荷を下げることや適切な質問順番の管理を可能にすることを目的とするオンライン質問応答システムを提案および実装する。ここでは、受講生が TA に質問するハードルを下げるため、TA に対して通話やチャットで直接質問をするのではなく、システムに対して具体的な質問を投げかけ、受講生のタイミングではなく、TA のタイミングで受講生を呼び、質問の対応を行う。また、TA には、質問をシステム上で事前にチェックできるようにし、自身が対応可能である場合にシステム経由で受講生を呼び出し、質問対

応することによって、精神的負荷も下げることを狙った仕組みにする。さらに、受講生が TA の対応を待っている間に自己解決した場合は質問を取り下げることができるなど、オンラインでの質問待ちを円滑にする仕組みを構築する。

以上のことにより、オンライン講義で受講生が TA に質問するハードルが上がってしまう問題を解決するとともに、TA の得手不得手を配慮しながらの質問対応を可能にし、さらには順番が来たときには受講生に通知を行うことによって気付かせ、講義を円滑にすることができるかと期待される。

本研究では、提案システムを askTA として実装するとともに、オンラインのプログラミング演習講義で実運用を行うことにより、その有用性を検証する。また、実運用から可能性や問題点などについても明らかにする。

本研究の貢献は、オンラインプログラミング演習講義での質問対応において、実運用に耐えうるシステムを提案し、長期的運用によりその有用性を示したことである。

2 関連研究

2.1 プログラミング指導支援に関する研究

大学におけるプログラミング演習型講義の支援に関する研究として我々 [18] は、これまでに TA と学生の抽象的思考に着目した指導を楽にするための手法を提案し支援を行ってきた。しかし、これまでの研究は対面講義を想定した TA と学生を支援するものであり、オンライン講義についての検討は不十分である。また、井垣ら [12] は、課題に対する取り組み速度などを利用し、サポートを必要とする学生を早期に検出できるシステムを提案し、実際の演習講義で 84% の精度でサポートが必要な学生の発見ができたとしている。市村ら [11] も、学生の操作やエラーログから、講義中につまずいている学生を早期に発見するとともに、共通する問題点を明らかにするシステムを提案し、実験で学生の手が挙がる前にかかえている問題に気づき、課題のヒントを出せたとしている。さらに加藤ら [14] は、模範解答との比較により、問題を抱える学生の学習状況を教員がリアルタイムに把握できるプログラ

表 1 対面講義とオンライン講義における主な課題

	対面講義	オンライン講義
TA の課題	苦手な分野だと解決に時間を浪費	TA 間での情報共有が困難
受講生の課題	初歩的な質問をするのが恥ずかしい	閉鎖的な通話環境への入室が困難
共通の課題	質問順番の管理が TA の記憶に依存	質問順番の管理が困難

ミング演習向け授業システムを提案し、進度の遅れている学生の検出とその学生の作業履歴の提示が可能で手法が有効であるとしている。

システムが自動的にアドバイスやフィードバックを行うことで講師や TA の負担軽減を目指す研究も存在している。Hattori ら [10] は、プログラムのコンパイル時にコンパイラが静的なエラーを吐くだけでなく、その先のユーザや状況も考慮した易しいアドバイスを享受する教育指向 Java コンパイラを提案しており、教師や TA の負担軽減に繋がるとしている。Marceau ら [17] も、初学者向けの丁寧なエラーメッセージを提示するために、エラーを修正するときの編集履歴を分析し活用する手法を提案しており、初学者の助けになる可能性について述べている。Petcha [27] は、e ラーニング上で、学生が提出したプログラムを自動で採点する機能に加えて、TA の代わりとなるようにプログラムのエラーパターンやフィードバックを提示する機能を提案している。この研究では一定の効果が示せたが、人間の TA には劣るため TA が対応できない状況に活用できるとしている。ArTEMiS [15] は、大規模な人数で行う演習講義においてオンラインエディタから提出された学生のプログラムを自動的に評価しフィードバックを返す目的で提案されている。その結果、初学者に適しており、徐々に学生のスキルを改善可能なことから TA の労力削減と学生の学習体験の向上ができるとしている。

プログラミングのコースの脱落を防ぐために学生が学び始めた初期のデータから最終成績や脱落を予測する研究も行われている。Munson ら [22] は、学び始めた 3 週間のプログラムにかかる時間やコンパイル時のエラー割合などと最終成績において相関があることを示している。また、Vihavainen [34] も、プログラムの編集履歴からインデントのミスや提出にかかった時間などからコースに合格できるかを 78% の精度

で予測している。さらに、Ahadi ら [1] は、学生がプログラミングを学習し始めた 1 週間のソースコード編集履歴から高い精度で成績が良くなる学生とそうでない学生の予測を行っている。

これらは、学生の理解状況を把握したり、TA の指導を楽にしたりするものであり、本研究とは別のアプローチで講義を良くしようとしている。これらの研究と本研究をシステムを運用する方法やシステムが学生に通知するエラーメッセージ、指導方法の点などにおいて組み合わせることで、講義をよりよくすることが可能になる。

2.2 プログラミングのリアルタイム共有と指導に関する研究

Goldman ら [7] は、2011 年にブラウザでリアルタイムにプログラムの共有と実行を備えた Collabode を提案し、アウトソーシング、ペアプログラミング、リモートクラスの 3 つの場面での運用について報告している。近年では、Google Colaboratory [8] や Visual Studio Live Share [20] などによって、リアルタイムにプログラムと実行環境を他者と共有することは容易になったが、プログラミング教育とその指導の観点からは不十分である。そのため、リアルタイムにプログラムを共有しながら指導に活かす研究が多くされている。Vandeventer [33] は、学生の進捗状況をリアルタイムで教師が観察できることやプログラム内の指定の箇所をリンクとしてメッセージを教師と共有できること、ログを再生できることなど、大学のプログラミング教育を支援する目的で作成されたシステムを提案している。また、Miller [21] らは、シンプルな操作ででき、かんたんに様々な OS にインストールが可能で、ステップ実行や変数のデータ構造を可視化可能なデバッガを備えた jGRASP を提案している。講義で使用した結果、デバッガが TA の支援になり、さ

らに学生もデバッガの使い方を習得したとしている。

学生同士のチームでプログラムを共有しながら課題に取り組むことを目的とした研究もある。Park ら [26] は、ピアアセスメントと呼ばれる協調学習の評価手段においてコードの編集履歴を可視化するシステムを提案し、それを見ることによって自他ともにプログラム構造や意図の理解が深まることを明らかにしている。加えて、動画再生のようにプログラミング環境上でプログラミングの講義を受けることができ、再生、停止、シーク、書き換えが可能なシステム [25] も提案している。ubrani ら [6] は、学生 2 名がそれぞれ作成しているプログラムの共有とチャット機能を設けた手法を提案し実装することによって、機能の有用性とコミュニケーションに必要な要件を明らかにしている。また、Codechella [9] は、プログラムをチュートや友達と一緒に書き、実行の変化について対話することが重要だとして、オンラインでプログラムをリアルタイムで共有しつつ変数の可視化が可能なシステムを提案している。9 ヶ月間の公開で 40 カ国 296 都市の人々が 299 回使用し、その 57% が複数都市間での共有であったと報告している。

以上のように、プログラミングをリアルタイムで共有しつつ指導を行う研究がされているが、本研究では、オンラインで TA に質問し、TA がリアルタイムで回答することを前提としたシステムを提案する。

2.3 ヒューマンリソースに関する研究

本研究では、質問したい受講生と TA とをマッチングする手法を提案する。この関係性は、予約受け付けやクレーム等の対応をしてほしい顧客と企業のコールセンターとの関係性に類似しているため、本研究の手法を明確化するために、コールセンターを題材とした人と人が関係するヒューマンリソースに関する研究について触れる。

コールセンターは顧客が電話をかける人数によって業務量が変化するが、企業はコストを減らす目的で適したオペレータの人数を配置したいと考えている。そのため、適切なオペレータの人数を導くために業務量を予測する研究が行われている。

Chuchual ら [5] は、コールセンターで電話を受け

るオペレーターと電話をかけるオペレーターの担当の割り当て方に着目したモデルを構築し、効率性とサービス品質に及ぼす影響を明らかにしている。また Athanassios ら [2] は、コールセンターには不確実な要素が多くシミュレーションが困難であることや多くの分析モデルが複雑に関係していることによる難しさについて述べている。

伊藤 [13] は、コールセンターに求められる品質には、顧客の待ち時間を減らし、適切なスキルを持ったオペレーターが的確に対応することが必要であると考察し、予測することの重要性を述べている。さらに、過去の実績データから日別や時間常別やテレビ CM などとも考慮した業務量を予測する手法を提案している。また、田原ら [31] は、コールセンターの過去 5 年分のデータに対して、日付や祝祭日かどうかなどの基本属性をもとに予測モデルを構築し予測誤差が減少することを確認し、手法の有用性を確認している。

関戸 [29] は、サービスサイエンスに基づいて、機器の保守に関するコールセンターのサービス改革を 6 つのステップを用いて行ったことを報告している。その中で、業務をプロセスに分解し、顧客にプロセスごとに期待度と満足度を調査しそのギャップが大きいプロセスを探すことで顧客が満足していないプロセスの発見を行っている。それによると、顧客が不満に感じていることはコールセンターより先の現場対応や進捗報告であり、コールセンターのオペレーターに対する不満はそれほど悪くなかったとしている。また、田口 [30] は、保険に関するコールセンターの顧客にコールセンターの不満について複数回答のアンケートを行っている。その中で、上位 2 つは「待ち時間が長い」が 59.4%、「話中が多い」が 32.9% となっており、オペレーターの人数不足がサービスやサービス品質に大きな影響を与えているとしている。

一方、玉川ら [32] は、コールセンターの離職率が高いことを問題に、音声感情データから離職懸念を抱えるオペレータを特定し、適切なケアを行うことを目指した研究を行っている。

以上のように、コールセンターを題材にしたヒューマンリソースに関する研究が多くなされている。

本研究では、受講生の人数と TA の人数が固定され

ているため業務量の予測は行わないが、業務量を定量化する際の参考に使用することができる。また、顧客の満足度に関して、待ち時間を少なくし、すぐ対応できることが望ましいのは確かであるが、満足度の向上に最も重要なのは質問した内容の解決であると考えられ、本研究では TA が丁寧に指導できる環境を構築することに重きを置く。さらにオペレーターに対して離職を防ぐためのケアが重要なことと同様に、TA が無理せずに業務を行える手法の提案を行う。

2.4 コミュニケーションの消極性に関する研究

栗原ら[16]は、消極的な人でも快適に使える情報技術のデザインを「消極性デザイン」と呼んでおり、研究会活動も行っている。また西田ら[24]は、実名と匿名の長所を併せ持つ傘連判状を用いたコミュニケーションプロトコルを提案しており、意見に圧力のかかる状況でも発言が可能であることなどを報告している。さらに、学会での交流を促進する目的で、近くになりたい人の希望を反映した席決めシステム[23]も提案し、実践的な運用を行っている。

TA は、受講生にとって同じ大学の先輩という目上の立場であることや、初歩的な質問をしただけで恥ずかしいというような受講生が消極的になりやすい環境であり、質問したくてもできない受講生やわからないことを放置する受講生がいると考えられる。また、TA 自身が元々内気な性格の場合に受講生に適切で十分な指導ができないことも考えられる。さらにオンライン環境では、よりコミュニケーションが取りにくくなるため、大きな問題となりうる。本研究もこれらの研究のように消極的になってしまう受講生や、もともと内気な性格の TA がオンライン環境でも快適に使い、質問をスムーズにやりとりできる手法を提案システムとして実装する。

2.5 マッチングに関する研究

教師と学生とのマッチングに関する研究について述べる。Chetty[4]らは、プログラミングコースを履修している学生が好む動画視聴学習や体験学習などの学習スタイルと教師の指導スタイルの関係を調査している。その結果、学習スタイルと講師の指導ス

イルとの間には学生の成績に有意な影響があることを明らかにしている。また、Awla[3]は、自然言語習得に関して認知的や人格的、感覚的に学生自身の学び方と教師の教え方が一致することで、学びが楽しく効果的になると結論づけている。

本研究でも TA と受講生のマッチングを行うため、プログラムを図示するのが得意な TA と図で理解したい受講生をマッチングや噛み砕いた話し方が上手な TA と言葉で解説してほしいと思っている受講生をマッチングするなど、個々の特徴を考慮したマッチングを行うことが受講生の習得や成績向上の観点から望ましいと言える。しかし、本研究では TA が限られた人数である状況を想定しているため個々の特徴を考慮したマッチングは実施せず、TA が受講生の質問を閲覧して選ぶことによってマッチングを行う。

3 提案手法

先述の通り、オンラインでは、受講生の質問ハードルはより上がると考えられる。また、TA も質問に対応する精神的負荷も高まると考えられる。さらに、こうした質問を適切にさばくための順番待ちなどの仕組みが重要になってくる。そこで、システムの実現にあたり、質問するハードルを下げる仕組みと、質問対応のハードルを下げる仕組み、質問の順番待ちを円滑に行うための仕組みについて述べる。

3.1 質問するハードルを下げる仕組み

オンライン上で実施されている演習講義では、TA などによる質問対応用にコミュニケーションスペース(質問部屋)を用意することがよく行われている。受講生は、質問するためにはそうしたスペースに入室する必要があるが、対面環境とは異なり、オンライン環境では部屋の中に誰が居るのか、どんな状況にあるのか、何を喋っているかなどを覗き見ることができないサービスが多く、入室のハードルがあがり、結果的に質問するハードルもより高くなってしまふ。また、質問のハードルを下げる方法として、受講生からの質問を Web フォームやチャットなどで受け付け、順次対応する方法も考えられるが、質問が多数ある場合に、誰がどれに対応するかの割り当てが難しく、また質

問対応で別システムが必要となるなどの問題がある。実際、1年間オンライン講義を受講してきた今回の運用の対象者ではない大学2年生41人を対象としたインタビューにおいてオンライン演習講義における難しさとして質問できず苦戦したことがあげられていた。

そこで、質問するハードルが高い問題に対して気軽に質問できるように、受講生がシステムに質問を投稿し、TAが受講生を呼び出す形式にすることで解決する。TAに直接質問するのではなく、システムに質問の投稿を行い、投稿された質問を見たTAに呼んでもらうことで、「TAに呼ばれたから質問できる」と受講生の立場を変えることで質問ハードルを下げることが可能になる。

3.2 質問対応のハードルを下げる仕組み

TAの業務は、どんな受講生の質問に対しても回答することであるが、TA個人にもレベル差や分野の得手不得手があり、質問の内容を聞くまでTA自身が答えられる質問かどうか分からない。もし不得意な分野の質問を受講生から受けた場合、回答に時間がかかることによるプレッシャーが焦りにつながり、精神的負荷が高まるという問題が発生する。

そこで、TAが質問対応の前に、質問の内容をある程度把握できるようにし、TA自身が回答できない内容であれば、その質問を別のTAに頼むことができるようにする。こうすることで、必ず質問が来たら対応しないといけないTAのプレッシャーを軽減することができる。

3.3 順番待ちを円滑に行うための仕組み

受講生からの質問が増加すると、質問の順番待ち行列が発生し、その行列を適切に管理する必要がある。また、受講生によっては、TAが対応するのを待っている間に自己解決できる可能性もある。ここで、自己解決したら順番を抜けられるようにすることは二度手間をなくすためにも重要である。

そこで、システムで質問された順番を管理し、順番が回ってきた受講生に対して、通知を行うことで質問の場への誘導を促す。また受講生の意思で順番から抜けられるようにする。これにより、受講生が自分の

質問順番に気付かない問題、TAが受講生に連絡をする手間や通知が飛ばない問題、自己解決しているのに順番を抜けられない問題を解決することができる。

4 askTA

提案手法を講義において実運用することを目指し、WebサービスaskTAとして実装を行い、演習講義で利用可能な質問応答システムを提案する。なお、受講生やTAを識別するため、大学発行のメールアドレスによる認証システムを使い、ログインを行ってもらったようにした。

4.1 実装

質問のハードルを下げつつ、事前にどのような質問なのかを把握可能とするため、質問投稿フォーム(図1)では、質問する課題の番号、質問の種類、質問の概要、できたところまでのプログラムという入力項目を設け、システムへの質問の投稿を可能にした。なお、質問の種類は、「ヒント」「ヘルプ」「問題文に関する質問」「採点に関する質問」「その他」の5つを用意し、選択式にした。質問内容によっては、できたところまでのプログラムの内容とは関係がない質問がされる場合も想定されるため、ソースコードの欄は省略可能とした。なお、運用の都合上、受講生を所属するクラスで分ける必要があったため、図1ではクラス(組)も選択するようになっている。

ダッシュボード(図2)では、投稿された質問の状況を一覧で見ることができ、受講生が他の受講生の質問の概要を見られるようにした。1つの質問には、質問の種類、質問者の名前、対応したTAの名前、質問の概要という情報をもたせ、質問のステータスは、「TA待ち」「TA対応中」「TA終了」「TA保留」「学生キャンセル」の5つを設けることで、質問対応状況を可視化した。質問のひとつひとつには、英数10桁の質問IDが割り当てられており、質問IDの表示はTAのみに行った。

TAが受講生の質問に対応する際には、受講生が書きかけているプログラムと口頭での会話による質問の詳細を把握することが重要である。そこで質問ごとにシェアエディタ(図3)を生成し、TAと受講生が

質問投稿フォーム

組(floor)

組を選択してください

質問をしたい課題

質問をしたい課題を選択してください

質問の種類

質問の種類を選択してください

質問の概要(1行程度)

例: 画面が黒くなる。(エラー文あると良い)

ソースコード(できたところまで良い/省略可)

```
void setup()
{
```

キャンセル 質問を送信

図 1 受講生が課題番号や質問の種類, 概要, プログラムを入力して質問を投稿するフォーム



図 2 投稿された質問の状況や内容が一目で分かるダッシュボード (個人氏名はモザイク加工)

同じ URL(質問 ID) にアクセスすることで, プログラムの共有と共同編集, 実行, 音声通話を可能とした. また, この画面上で質問の対応開始や対応終了などの TA による質問ステータスの更新や, 受講生による質問の取り下げ (図 4) を設定可能とした. なお, 対応中の TA の名前や受講生の名前, 互いのカーソル位置も表示することで, 質問対応を容易化している. また, 受講生による不正行為防止のため, 受講生は他者のシェアエディタにアクセスしてプログラムなどの質問の中身を見ることができないようにした.

図 5 は, 受講生に質問の順番が回ってきたことを知らせる通知である. しかし, この単純な通知では課題に取り組んでおり気づかない可能性があるため, 5 秒程度の効果音も同時に鳴らすことで順番が回ってきたことの気づきを促した.

提案システムでは, 我々の学科のプログラミング教

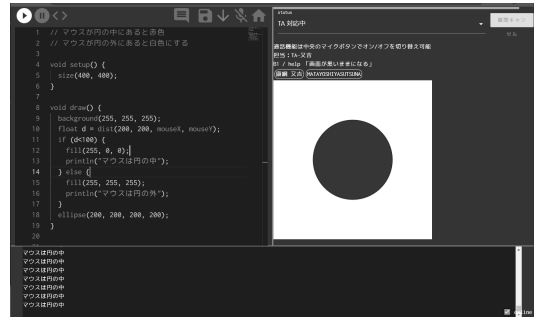


図 3 受講生と TA がプログラムの共有や実行を音声通話をしながら質問を解決するシェアエディタ



図 4 受講生が質問を取り下げることができる質問キャンセルボタン

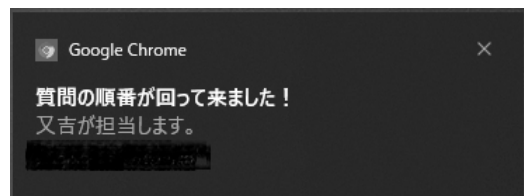


図 5 受講生に順番が来たことを伝えるトースト通知

育に Processing が採用されていることから, Web 上で実行を行える Processing.js を Processing と同じような挙動を行うように改良して実現した. なお, セキュリティの観点から Web 上で実行する前にサーバ側の Processing でコンパイルし, 実行可能かどうかについて確認を行った. また, サーバ側のコンパイルに失敗した場合は, エラー行数とエラー文に加え日本語に翻訳したエラー文も表示した.

多様な UI がある中でも操作がわかりやすいように汎用的なアイコンを使用したり, Processing と同様のショートカットキーを使用できるようにするなど使い勝手のデザインにもこだわった.

4.2 利用方法

図 6 に受講生と TA がシステムを利用する際の画

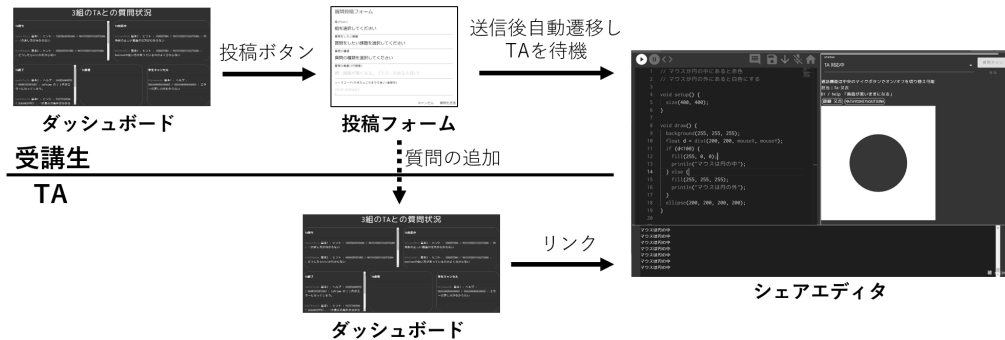


図6 受講生とTAの画面遷移の概要

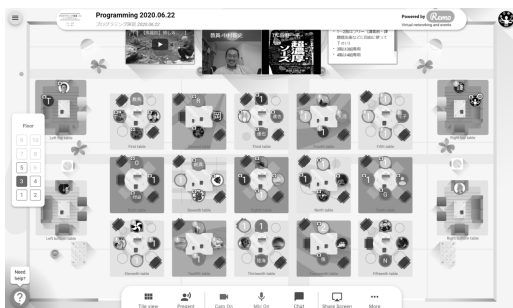


図7 中央のテーブルに受講生が座り、周りにTAが質問の待機している講義中のRemo Conferenceの様子

面遷移を表す。

受講生は、システムにログインしたうえで、質問投稿フォームより解決したい質問を投稿する。質問を投稿した後は、自動的にシェアエディタに遷移し、TAが対応可能となるまで、ブラウザのタブとしてシステムを開いておいて順番待ちをすることになる。

TAはシステムにログインすると、ダッシュボードを確認でき、質問対応待ちや対応済みなどといった質問の状況を把握することができる。また、リンクになっている質問IDをクリックすることでシェアエディタ画面に移り、質問の概要とプログラムを確認できる。ここで、質問やプログラムの内容から対応可能であると判断した場合は、ステータスを「TA 対応中」に変更することで、質問対応を始めることができる。対応不可だと判断した場合は、他の質問をチェックしていくことになる。

TAが「TA 対応中」にステータスを変更すると、受

講生には通知が飛び、呼び出しが行われる。その後、シェアエディタ上でやり取りを行うことにより質問の解決を行う。なお、受講生が質問の順番までに自己解決できた場合は、シェアエディタ右上の「質問キャンセル」ボタンをクリックすることで質問を取り下げることができる。

5 運用と分析

5.1 運用

明治大学総合数理学部先端メディアサイエンス学科のプログラミング演習Ⅰの講義において、2020年7月1日から27日までの毎週月曜日と水曜日に全8回の運用を行った。プログラミング演習Ⅰの履修者は学部1年生と再履修者による111名で、TAは主著を含む、大学院生9名だった。オンライン講義のためのツールとしては、受講生が複数あるテーブルを自由に移動ができるRemo Conference[28]を使用し講義を実施していた(図7)。Remo Conference上では、テーブル内での音声通話やビデオ通話、画面共有、チャットを行うことができる。また、テーブル内での通話を止め、講師が全員にWebカメラ映像や画面共有などをブロードキャスト配信することができる。

講義(100分の2コマ)では、typing.run[19]や予習資料などをもとに予習してきたことを前提に、スライドを用いて講師が説明を行い、講義時間終了までに提出する必要がある課題を課した。また受講生は、課題解答中はRemo Conference上で指定されたテーブルで作業を行い、簡単な質問は同じテーブルの受講生で助け合い、解決できない質問についてaskTAで質問

してもらった。TA は、受講生の質問対応と提出された課題のチェックと 2 つの業務を行った。

なお、Remo Conference の制約上 60 人程度ずつに分ける必要があったため、受講生を 2 クラスに分け、TA も各クラス 3 人配置し、質問状況に応じて、TA はクラス間の移動を行った。9 名いる TA のうち 6 名は質問対応を行い、残りの 3 名は提出された課題のチェックを中心に行っていた。しかし、質問を待っている受講生が多くなった場合は、課題のチェックを行っていた TA が応援として質問対応を行うようにし、逆に質問を待っている受講生が少なくなった場合は、課題のチェックを手伝うというような柔軟な運用を行った。

運用の 4 回目までを終えた時点で、TA が受講生の質問に対応している時間が長くなりすぎる傾向が見られ、質問対応が滞ってしまう問題が生じた。そのため、5 回目以降の運用では、1 回の質問当たり 10 分を目処とするように TA に口頭で依頼するとともに、シェアエディタの右上部に 10 分間を測る水色のバーを設けた。なお、この措置に強制力はなく、対応を行う時間は TA に任せた。

5.2 利用状況の結果と分析

運用期間中にされた質問の回数は 367 回だった。そのうち 15 回は、TA が対応する前に受講生によってキャンセルされていた。質問した人と質問回数の関係のグラフを図 8 に示す。1 回以上システムを使って質問した人は 61 名おり、そのうち 5 回以上質問した人は、39.3% にあたる 26 人だった。一番質問回数が多い人は、28 回質問していた。図 9 は、受講生が質問を投稿した時間と TA が「対応中」にステータスを変更した時間の差を取ることで、受講生の質問に TA が反応するまでの待ち時間 (分) の分布を求めたものである。この結果より、51.6% の質問は、1 分以内に TA が対応していたことが分かる。図 10 は、TA が「対応中」と「対応終了」にステータスを変更した時間の差を取り、TA が受講生の質問に対応した時間 (分) の分布を求めたものである。この結果より、対応時間が 4 分台の質問がもっとも多かったことが分かる。また、ほとんどの質問は 20 分以内には対応

が終わっていることも分かる。なお、講義中の休憩時間 (10 分程度) や講師による補足解説で TA 業務が一時中断中もシステムを運用しており、図 9 では待ち時間、図 10 では対応時間にこれらの TA 業務が一時中断していた時間を含む質問があった。

図 11 は、横軸に講義時間、縦軸に質問のステータス (1: TA 待ち, 2: TA 対応, 3: TA 終了, 4: TA 保留, 5: 学生キャンセル) を取ることで、受講生の質問ごとにどのようなステータスの遷移を行ったかが時系列でわかるように全 8 講義分を可視化した結果である。横軸の時間は 7 月の日付と時間を表し、1 つの折れ線は受講生 1 名がした質問を表しており、マークは対応した TA ごとに決めた一意のマークをしている。受講生によって質問がキャンセルされた場合はマークを付けていない。また、塗りかされている時間帯は TA 業務が一時中断した時間帯である。TA によって休憩をとる時間が異なっているため、TA の休憩時間は TA 業務の一時中断した時間には含まれておらず、講師の補足解説による中断した時間だけを示している。これより、時間によっては多くの質問が異なる受講生から同時に送信されたり、比較的長時間 TA の指導を受けていたりしている様子が観察できる。また、TA が順番よく対応していることもわかる。なお、TA は講義ごとに学生からの質問に対応 (6 名) と課題のチェック (3 名) に分かれていたため、講義によって担当する TA が異なりマークの分布も日ごとに異なる。しかし、第 5 回目の運用に当たる 15 日の 16 時 35 分付近に、この日 1 本だけ上向き三角の TA が混ざっていることがわかる。これは、10 分前の 16 時 25 分付近に質問が集中し、待っている受講生が多くなったため、課題のチェックをしていた TA が一時的に応援として質問対応をしたことが読み取れる。

5.3 受講生アンケートの結果と分析

運用終了後にシステムを使った受講生にアンケートを行った。アンケートでは、54 名 (askTA を利用した受講生は 61 名) から回答を得た。表 2 はそのアンケートの結果である。なお、アンケートでは質問に対する 5 段階評価 (-2: 全く当てはまらない ~ 2: とても当てはまる) での回答と、自由記述で感想を求めた。

表 2 システムを使用した受講生のアンケート項目とその分布と平均

質問項目	評価値					平均値
	-2	-1	0	1	2	
使った満足度は高いですか？	0	0	3	14	32	1.58
TA に質問する抵抗が低くなりましたか？	0	3	17	14	16	0.91
プログラミング講義で今後も使いたいですか？	0	0	1	13	35	1.66

表 2 より、システムを使って満足度が下がった人や今後の利用はしたくない人はいないことが分かるうえ、回答者の半数以上について、質問への抵抗感を下げることができているのが分かる。自由記述のポジティブな意見として、「質問を端的に伝えられた」「オンライン講義での質問のしにくさの不安が解消された」「質問することのハードルがかなり下がった」などのように、我々が意図した意見が得られた。また、「プログラムの共有が役に立った」「プログラムを実行できるのが助かった」などの機能に関することも多かった。一方、ネガティブな意見では、「質問を投稿する時に書く概要がうまくまとめられなかった」「プログラムの共有だけでいいと思った」と言った意見も得られた。また、「TA のカーソルが目立つようにしてほしかった」「ボタンを押す間違えた」「実行結果がシステム上では違った」など、実装に関することも多かった。システムの使い勝手に関する意見は得られなかった。

さらに、システムを使わなかった受講生にも同様のアンケートと自由記述で askTA についての感想を求めた。アンケートでは、37 名 (askTA を利用しなかった受講生は 50 名) から回答を得た。「TA に質問する抵抗感が低くなったか」の問には、1 名が -1、33 名が 0、3 名が 1 と評価していた。また、「今後も使いたいか」の問には、1 名が -1、19 名が 0、11 名が 1、6 名が 2 と評価していた。自由記述では、「TA の力を借りなくても課題を解くことができたので利用しなかった」「掲示板のように回答が遅い代わりに気軽に聞ける場所が欲しかった」という感想があった。

5.4 TA アンケートの結果と分析

運用終了後に、システムを使った TA のうち、主著を除く 8 名にアンケートを行った。アンケートでは 5 段階 (-2: 全く当てはまらない~2: とても当てはまる) で評価してもらうとともに自由記述での感想を求めた。

アンケートの結果を表 3 に示す。この結果から、システムを使うことの満足度が高く、今後も使いたいと TA が感じていると言える。

また、自由記述のポジティブな意見としては「どれぐらいの人がどこで困っているのかが一覧でわかった」「質問状況とその内容をから溜まっている質問の量とその内容を確認できた」「挙手よりも質問に気づきやすい」という提案手法の質問をシステムに投稿することや共有することによるものと思われる意見や、「通知機能ですぐに受講生が来た」という通知機能によるもの、「受講生が修正する様子がカーソルで見える」「プログラムがリアルタイムに共有されるのでカーソルで指をさせる」「行番号が助かった」「通知機能でも来ない人に対して通話機能で話しかけられた」というシェアエディタの実装に関することも多いことが分かった。また、「askTA 単体で TA 業務が成り立つ」という意見もあった。一方、ネガティブな意見として、「TA の引き継ぎができないため、他の TA が指導した続きを教えるのがキツかった」「TA も自由に楽に編集できるので教えずぎてしまうことがあった」というオンライン特有の意見や、「TA 側にも通知機能が欲しかった」「図を使って説明したかった」「実行環境にバグがあった」という実装に関する意見もあった。また、「askTA に頼り切りの人がいた」という対面講義時と同様に TA に頼り切る受講生がいたこともわかった。

表 3 TA のアンケート項目とその分布と平均

質問項目	評価値					平均値
	-2	-1	0	1	2	
使った満足度は高いですか？	0	0	1	4	3	1.25
プログラミング講義で今後も使いたいですか？	0	0	1	1	6	1.63

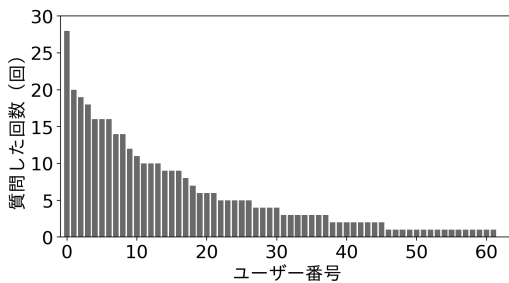


図 8 質問した人とその回数の関係

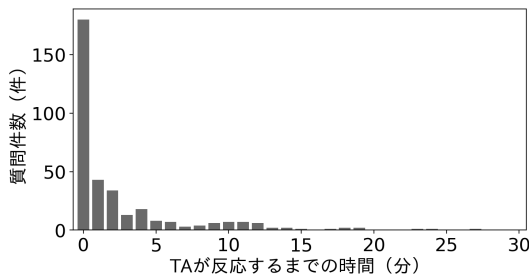


図 9 受講生の質問に TA が反応するまでの待ち時間 (分) の分布

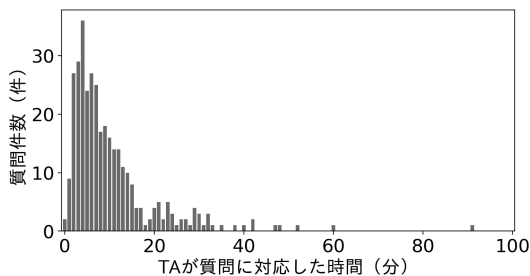


図 10 TA が受講生の質問に対応した時間 (分) の分布

ダッシュボードに掲載される質問の回答方法について詳しく記述を求めたところ、「基本的には上から順に対応した」「1 度対応し、再び質問する受講生には積極的に対応していた」といった一方で、「自分が

理解していない課題の質問は他の TA が対応したほうが良いと思ったため対応しなかった」「概要の文章が乱雑な質問には対応したくなかった」「対応に躊躇してただ眺めてるときもあった」「受講生の質問対応の担当ではないときにサポートで入ったが、難しそうな質問が来たら対応せずにサポートをやめた」といった感想があった。

6 考察

6.1 利用状況の考察

図 8 より、システムを利用した人は受講生の約半分であり、広く利用されていたことが分かる。同じ人が何回もシステムを使って質問していることに関しては、対面講義でもよく見られる現象であるが、TA がヒントを一度与えて、もう一度来るように指導したことも原因として考えられる。なお、質問回数が最も多かった受講生は再履修生であり、単位を取るために質問を繰り返していた可能性がある。

図 9 より、ほとんどの質問は 1 分以内に TA が対応するが、数分や数十分待つことになった質問もあったため、本手法の通知機能が活躍したと考えられる。なお、TA が質問対応を行うかの選択が可能であるが TA 全員が対応しなかった質問はなかった。

また図 10 より、質問の対応に長く時間かかることもあり、TA が指導の順番を把握するのは困難であったと言え、システムで順番を管理することで TA の負担を下げるのができた可能性がある。

図 11 より、多くの折れ線が 3 点 (1: TA 待ち, 2: TA 対応, 3: TA 終了) で表され、1 から 2 の長さは短く、傾きが急であり、2 から 3 の長さは長く、傾きが緩やかであることで、受講生は少ない待ち時間で TA に対応してもらえつつ、それ以上の長い時間をかけて質問対応している傾向が見える。また、TA

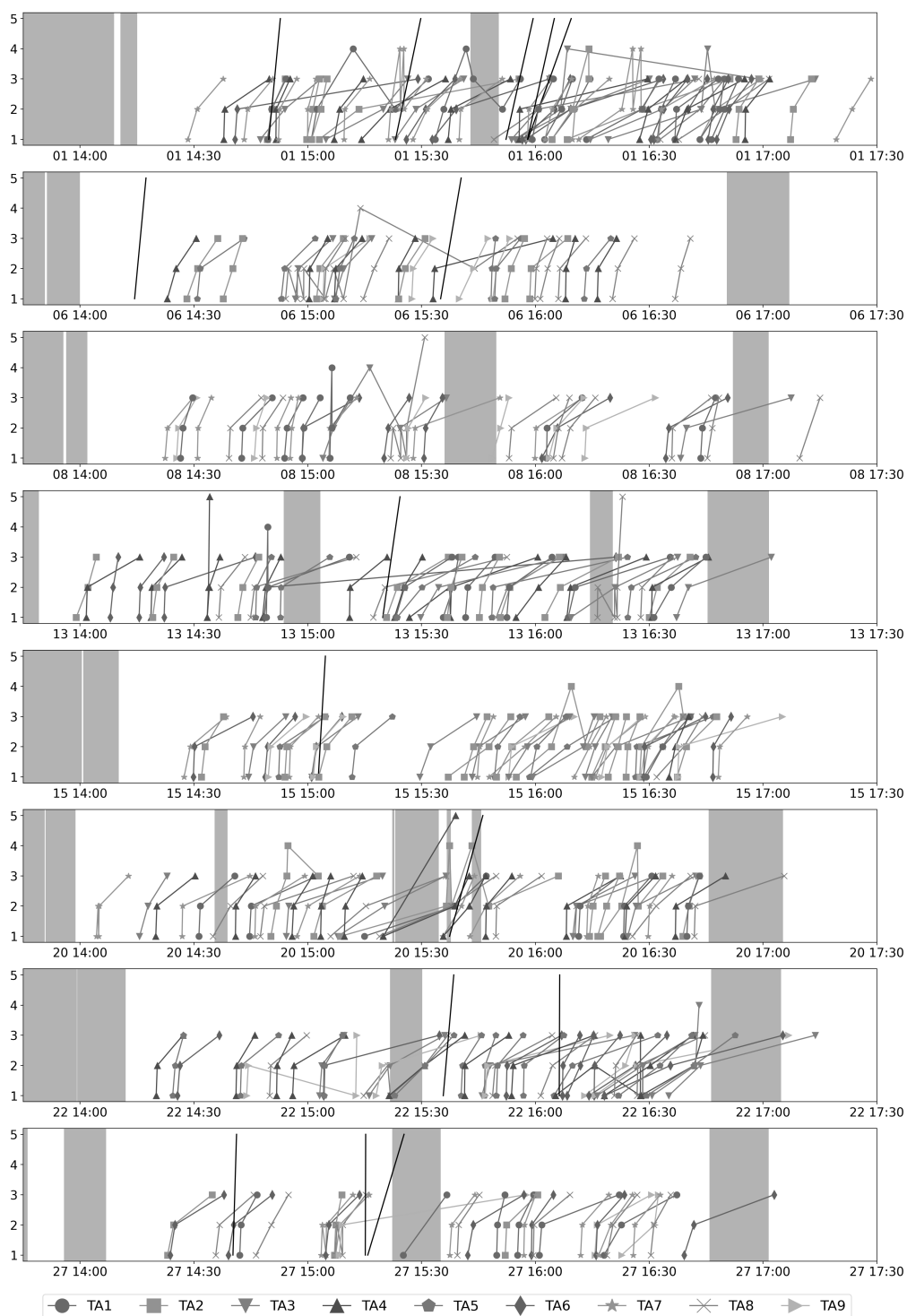


図 11 TA ごとに一意なマークで示された受講生の質問に対するステータス変化を縦軸に時間を横軸にとった可視化 (塗りの領域は TA 業務中断時間)

業務が一時中断したことを示す塗られた時間帯を横切るかたちで折れ線が伸びている質問は、TA 業務の再開と同時に対応終了ステータス（縦軸が 3）になっている質問が多くある。これは、TA 業務の中断中に行った講師の補足説明の中で受講生が TA にしていた質問が解決されたことによる現れであると考えられる。さらに、7 月 15 日の講義では、最初の 30 分以降、Remo Conference に接続できなくなり、最後まで復帰しないトラブルが発生した。そのため、図 11 の TA 業務中断時間を示す塗りのデータも欠損した。しかし、システムの機能である質問の投稿とステータス、及びシェアエディタのプログラム共有、音声通話より、特に大きな支障なく受講生は課題に取り組むことができ、TA もその時間に質問された 51 件について十分に質問対応を行うことができていた。以上より全体的に、講義によって質問数や密度の違いがあるものの、TA が忙しく受講生の質問にシステムを適切に用いながら対応していたことが読み取れる。

次に、受講生と TA の人数比について、考察を行う。運用では、質問したい受講生に対して質問をテーブルで相談し、解決できなかったらシステムを用いるように指導した上で、受講生の 10% 弱の人数の TA で十分に対応することができ、受講生と TA の双方から一定の評価を得た。TA の業務量は、受講生の質問数と質問の重さによって決まるため、課題の難易度によって TA の業務量が左右されると言える。最適な受講生と TA の人数比は課題の難易度に左右されるが、難易度が高く受講生の質問が多くなると予想される場合には、講師が適宜全体向けにヒントや解説をすることで TA の業務量を少なくすることが可能である。そのため、TA の人数を固定するのではなく、柔軟に受講生の質問状況を見ながら TA の人数を増やしたり、講師がヒントを出すといった対応が最善だと考えられる。

6.2 受講生アンケートの考察

表 2 の結果より、満足度と今後も使用したいの質問に対して、負の評価をした受講生はおらず、受講生からの評価は高いものであった。また TA に質問することの抵抗に関しては 0 を選択した受講生が最も

多かったが、正の評価をした受講生が倍に近い人数であったことから、TA に質問する抵抗感も低くすることが可能であったと言える。

自由記述で得られたポジティブな意見から、提案手法の質問するハードルを下げる効果が出ていると言える。一方、ネガティブな意見から、受講生が質問投稿時の概要の記述が難しいと感じていることがわかった。概要は、TA が対応可能であるかを判断するために重要な項目であるため、適切な記述が求められる。しかし、質問の概要には、「よくわからない」「だめなところがわからない」「2 行目がおかしい」「エラーが起こる」などといった、具体的な質問内容を記述しないものが一定数あった。これは、受講生に概要の書き方について指導しなかったことで起こったと考えられる。これについては、指導で「エラー文を含んでください」や「悩んでいることの主語（何が動かないのか）を可能な限り記述してください」などと運用前に丁寧に説明することで改善可能である。なお、適切に記述できていた受講生の質問の概要は、講師が、課題解決に際し、どういった問題が発生しているかをダッシュボードから容易に把握できるため、適切なタイミングで全体向けのヒントを提示することにつながっていた。

また、受講生が TA を待つ時間に関する自由記述が 1 つもなかったため、待ち時間に関する不満はなかったと考えられる。TA からの呼び出しを待っている間は、ブラウザのタブでシステムを開いておくだけで良いため、受講生自身が質問依頼を出しつつも、課題を解く手を止める必要がない点が良かったと言える。さらに、ダッシュボードで常に質問状況が受講生にも共有されるため、TA が来るまでの目安の時間がわかったことも待ち時間に関する不満がなかった原因として考えられる。

システムを利用しなかった受講生へのアンケート結果より、システムを使わなかった受講生は TA に質問することがないため利用しておらず、システムを通じて質問することが嫌だったりシステム自体が質問へのハードルとなり質問をしなかった受講生はほとんどいなかったと推測される。

6.3 TA アンケートの考察

表3の結果より、満足度と今後の使用に関して、負の評価をしたTAはおらず、各質問項目の平均も1を超える高いものとなっていることから、TAはシステムを好意的に捉えていたことが分かる。

自由記述からは、提案手法の順番待ちをしっかりと行えることや質問が事前に分かること、通知機能が評価されていることが分かった。さらに、カーソル位置の共有や通話機能などのシステムの細かい実装や些細な工夫が評価されていた。しかし、同じ受講生に対して同じTAが対応できないという問題が指摘された。対面講義では、TAが継続して同じ受講生を見ることができ、「ココまできたらまた呼んで」など、解法の流れを区切って教えることや質問意図の疎通が楽になるメリットがあるため、手を上げた受講生をTAが誰か判断し、継続で見たほうがいいのであれば、TA間で融通を利かせて指導を行うといった工夫がみられていた。一方、提案システムは受講生がTAを指定して呼ぶことはできないうえに、TA間コミュニケーションを促して、融通を利かせられたりするような設計を行っていなかった。これらから、質問したい受講生とTAをマッチングするだけではなく、マッチングまでのTA間でのコミュニケーションや、受講生とTAの指導中のハードルが下がったことによって生じる新たな指導方法に関する問題が発生していると考えられる。また、実行環境にバグがあったことや、TAも質問が来たことの通知がほしいなどのUIや機能に関する意見も得られていたが、これについては実装に反映し改善を行った。

ダッシュボードに掲載される質問の回答方法についての記述より、TAは業務のため質問が来たら順序よく対応していた一方で、他のTAに質問を回していたことや、躊躇して対応しなかったことがあることが明らかになった。このことから、本提案手法によりTAの質問対応のハードルを下げるができたと言える。

7 制限と応用

本研究で実現したaskTAは実用性を重視しており、オンライン型の演習講義において質問応答を問題な

くさばくことを主目的としていたため、細かい機能の妥当性については十分に評価できていない。また、2年間にわたりトラブル無く質問対応ができていたり、大学公式の授業評価アンケートの結果などから高い評価を得ていることはわかっているが、現時点で十分に効果を発揮できていない機能の存在や、どういった機能があるとより効果的になるのかなどが明らかになっていない。そこで今後は、本システムを様々な教育現場に広げていくため、こうした点についても明らかにしていく予定である。また、1つの演習講義でしか運用をしていないため、本手法が有用である講義形式の条件などが明らかになっていない。講義を行う講師によって重視する点が異なることや、大学によって受講生からの質問の傾向が異なることも考えられるため、複数の条件において、手法がどのような状況で有効であるかを明らかにする必要がある。また、質問を投稿する時に書く概要が難しいという意見に対して、我々が以前取り組んでいた、抽象的思考を用いた受講生とTAのプログラミング教育円滑化手法[18]を応用し活用することで、プログラムを思考する力を鍛え、自己解決を支援するとともに、TAが指導しやすくする手法を実現していく予定である。さらに1人のTAが複数人の受講生を一斉に指導する状況や、TA間での受講生の引き継ぎや継続指導がaskTAでは考慮されていないため、今後機能の改善を行う必要がある。実現したシステムを様々な大学で利用してもらうことにより、オンラインにおける学びを支援していきたい。

一方、askTAでは、質問対応がすべてオンラインで行われるため、受講生が記述した質問の概要やプログラム、さらには受講生とTAの音声通話のやり取りをすべて記録、録音することが可能となる。通常のTA業務を行いつつデータを収集することが可能であるため、講義中に起こる質問をさまざまな角度から分析ができ、1講義内で行われた受講生とTAの質問のやり取りの内容を俯瞰的に示す機能の実現や支援が必要な受講生の早期発見、TAの得意不得意を考慮したマッチングなども可能になる。さらに、蓄積することでTAに受講生への指導方法の提案や受講生のプログラムから質問内容を予想・回答する人工知能な

ど、より良い TA 業務に繋げることができるようになる。また、オンライン講義だけでなく、対面講義でも askTA を導入することで、順番待ちの効率化や共同編集と実行などによって、ハードル低下による受講生の質問数増加や TA の負担軽減、受講生との意思疎通の手助けが可能になる。さらに、askTA は演習型のプログラミング講義を対象としているが、他のプログラミング以外の講義にも応用することができ、質問投稿の仕組みと TA との音声通話だけでも TA の負担や受講生の質問を快適できる可能性がある。

8 まとめ

本研究では、大学のプログラミング演習型のオンライン講義において、受講生が TA へ質問するハードルが上昇してしまう問題と TA が精神的負荷を覚える問題、質問の順番待ちを円滑に行う必要がある問題に対して、質問をシステムに投稿し呼んでもらうことで、質問する心理的ハードルを下げるとともに、TA 側も事前に質問を把握することで精神的負荷を下げるができる手法を提案および実装し、実際の講義における運用でその有用性を明らかにした。その結果、ほとんどの質問は受講生が質問を投稿して 5 分以内に TA に対応され、対応時間については 20 分以内のものが多かった。これらから、受講生の質問ハードルを下げていたことや TA の精神的負荷を低減できていたこと、質問の順番待ちが円滑に行われていたことを明らかにした。また、受講生の困っていることを俯瞰できること、オンライン講義のためのシステムが落ちても対処できることなどの副次的な効果もあった。

実装とコミュニケーションデザインの 2 つの工夫により、実際のオンライン講義において提案システムで 367 回の質問がされ、TA と受講生をオンライン上でマッチングを行え、講義を円滑に行うことが可能であった。さらに、2021 年度では対面講義を主としながら希望する受講生や TA にオンライン講義でも実施するといったハイブリッド運用を行う状況においても十分に機能し活用することが可能であった。特に、対面講義において質問を行う学生と TA との距離が近くなり感染リスクを上げてしまうという問題について、質問に関するやり取りで askTA を活用するこ

とにより、感染リスクのない質問応答を可能としていた。こうした知見は、今後対面講義に戻ったとしても活用できるものであると考えられる。

謝辞 提案システムを講義で活用してくれた明治大学総合数理学部先端メディアサイエンス学科の 8 期生及び、対応してくださった TA のみなさまに感謝します。

参考文献

- [1] Ahadi, A., Lister, R., Haapala, H., and Vihavainen, A.: Exploring Machine Learning Methods to Automatically Identify Students in Need of Assistance, in *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ACM Press, 2015, pp. 121–130.
- [2] Avramidis, A. N. and L'Ecuyer, P.: Modeling and simulation of call centers, in *Proceedings of the Winter Simulation Conference*, IEEE, 2005, pp. 144–152.
- [3] Awla, H.: Learning Styles and Their Relation to Teaching Styles, *International Journal of Language and Linguistics*, Science Publishing Group, 2014, pp. 241–245.
- [4] Chetty, N., Handayani, L., Sahabudin, N., Ali, Z., Hamzah, N., and Kasim, S.: Learning styles and teaching styles determine students' academic performances, in *International Journal of Evaluation and Research in Education (IJERE)*, Institute of Advanced Engineering and Science, 2019, pp. 610–615.
- [5] Chuchual, P., Chongpravatisakul, N., Kusolmanomai, T., and Komolavanij, S.: Inbound and outbound calls assignment for an efficient call center, in *2010 7th International Conference on Service Systems and Service Management*, IEEE, 2010, pp. 1–4.
- [6] Čubranić, D. and Storey, M. A. D.: Collaboration Support for Novice Team Programming, in *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work*, ACM Press, 2005, pp. 136–139.
- [7] Goldman, M., Little, G., and Miller, R. C.: Colabode: Collaborative Coding in the Browser, in *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, ACM Press, 2011, pp. 65–68.
- [8] Google: Colaboratory - Google Colab. <https://colab.research.google.com/> Accessed: 2021-02-25.
- [9] Guo, P. J., White, J., and Zanelatto, R.: Codechella: Multi-user program visualizations for real-time tutoring and collaborative learning, in *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2015,

- pp. 79–87.
- [10] Hattori, S. and Kameda, H.: knowledge-based Compiler with e-TA for Software Engineering Education, in *Proceedings of the 9th Joint Conference on Knowledge-Based Software Engineering*, Springer, 2010, pp. 265–278.
 - [11] 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.
 - [12] 井垣宏, 齊藤俊, 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.
 - [13] 伊藤稔: コールセンターにおけるインバウンド予測 (特集: データ分析), *Unisys 技報*, Vol. 25, No. 3 (2005), pp. 293–304.
 - [14] 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.
 - [15] Krusche, S. and Seitz, A.: ArTEMiS: An Automatic Assessment Management System for Interactive Learning, in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ACM Press, 2018, pp. 284–289.
 - [16] 栗原一貴, 西田健志, 濱崎雅弘, 築瀬洋平, 渡邊恵太: 消極性デザイン宣言 消極的な人よ、声を上げよ。……いや、上げなくてよい。、ビー・エヌ・エヌ新社, 2016.
 - [17] Marceau, G., Fisler, K., and Krishnamurthi, S.: Measuring the Effectiveness of Error Messages Designed for Novice Programmers, in *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, ACM Press, 2011, pp. 499–504.
 - [18] Matayoshi, Y. and Nakamura, S.: Abstract Thinking Description System for Programming Education Facilitation, *Learning and Collaboration Technologies. Human and Technology Ecosystems*, Zaphiris, P. and Ioannou, A.(eds.), Vol. Lecture Notes in Computer Science 12206, Springer International Publishing, 2020, pp. 76–92.
 - [19] 又吉康綱, 中村聡史: typing.run: 初学者のプログラミング学習を支援するプログラムタイピングシステムの提案と実践, 研究報告ヒューマンコンピュータインタラクション (HCI), Vol. 2020-HCI-189, No. 1 (2020), pp. 1–8.
 - [20] Microsoft: Visual Studio Live Share — Visual Studio - Visual Studio. <https://visualstudio.microsoft.com/ja/services/live-share/> Accessed: 2021-02-25.
 - [21] Miller, A., Reges, S., and Obourn, A.: JGRASP: A Simple, Visual, Intuitive Programming Environment for CS1 and CS2, *ACM Inroads*, Vol. 8, No. 4 (2017), pp. 53–58.
 - [22] Munson, J. P. and Zitovsky, J. P.: Models for Early Identification of Struggling Novice Programmers, in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ACM Press, 2018, pp. 699–704.
 - [23] Nishida, T.: Designing Social Interaction Support System with Shyness in Mind, in *Proceedings of the 2018 ACM Conference on Supporting Groupwork*, ACM Press, 2018, pp. 140–144.
 - [24] 西田健志, 五十嵐健夫: 傘連判状を採り入れたコミュニケーションプロトコル, 情報処理学会論文誌, Vol. 51, No. 1 (2010), pp. 45–53.
 - [25] Park, J., Park, Y. H., Kim, J., Cha, J., Kim, S., and Oh, A.: Elicast: Embedding Interactive Exercises in Instructional Programming Screencasts, in *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, ACM Press, 2018, pp. 1–10.
 - [26] Park, J., Park, Y. H., Kim, S., and Oh, A.: Eliph: Effective Visualization of Code History for Peer Assessment in Programming Education, in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ACM Press, 2017, pp. 458–467.
 - [27] Queirós, R. A. P. and Leal, J. P.: PETCHA: A Programming Exercises Teaching Assistant, in *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, ACM Press, 2012, pp. 192–197.
 - [28] Remo: Remo.co Virtual Events and Office space - Conference - A Virtual Happy Hour for Remote Teams. <https://remo.co/conference/> Accessed: 2021-02-25.
 - [29] 関戸隆明: サービスサイエンスの実践による顧客満足度と生産性の向上, *デジタルプラクティス*, Vol. 1, No. 1 (2010), pp. 21–28.
 - [30] 田口浩: コンタクトセンタにおける社員満足度と顧客満足度の関係性について, *デジタルプラクティス*, Vol. 9, No. 2 (2018), pp. 388–413.
 - [31] 田原琢士, 王軼謳, 山浦佑介, 大西健司: コールセンターを対象とした業務量予測に関する研究, 人工知能学会全国大会論文集, Vol. JSAI2017(2017), pp. 1–4.
 - [32] 玉川諒, 有次正義: 音声感情データ解析によるコールセンター従事者の離職懸念発見手法の提案, 第 12 回データ工学と情報マネジメントに関するフォーラム論文集, (2020), pp. 1–5.
 - [33] Vandeventer, J. and Barbour, B.: CodeWave: A Real-Time, Collaborative IDE for Enhanced Learning in Computer Science, in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, ACM Press, 2012, pp. 75–80.
 - [34] Vihavainen, A.: Predicting Students' Performance in an Introductory Programming Course Using Data from Students' Own Programming Process, in *2013 IEEE 13th International Conference on Advanced Learning Technologies*, IEEE, 2013, pp. 498–499.

**又吉 康綱**

1996 年生。2021 年明治大学大学院
先端数理科学研究科博士前期課程卒
業。現在、株式会社日本経済新聞社
にてソフトウェア開発に従事。オン

ラインでのプログラミング教育支援システムや、機械
学習によって推定されたダンスモーションの誤り訂
正システム、手書き文字美化システム等の研究活動に
従事。修士 (工学)。

**中村 聡史**

1976 年生。2004 年大阪大学大学院
工学研究科博士後期課程修了。同年、
独立行政法人情報通信研究機構専攻
研究員。2006 年京都大学大学院情報

学研究科特任助手、2009 年同特定准教授、2013 年明
治大学総合数理学部先端メディアサイエンス学科准教
授、2018 年同教授、現在に至る。サーチとインタラク
ションや、ネタバレ防止技術、平均手書き文字等の研
究活動に従事。情報処理学会等の会員。博士 (工学)。