

askTA: 消極的な受講生でも質問可能なオンライン演習講義支援システム

又吉 康綱* 中村 聡史*

概要. 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。

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[1]` 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。

ここで対面型の講義では、質問したい受講生が手を挙げて TA を呼び、受講生は画面を一緒に見ながら質問を解決し、TA は資料を指しながら指導をすることになる。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、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、オンライン上に質問部屋を用意して TA に待機してもらい、その部屋に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し、その結果精神的な負荷がかかるといった問題がある。

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 の精神的負荷を下げるのが可能なオンライン上の質問応答システムの実現を目指す。ここでは、TA に質問するハードルを下げるため、TA に対して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、受講生自身のタイミングではなく、TA のタイミングで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。

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。

2 関連研究

大学におけるプログラミング演習授業を支援する研究は多く存在する。井垣ら[2]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受講生を早期に検出できるシステムを提案している。また市村ら[3]も、受講生の操作やエラーログから、授業中につまづいている受講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ている。さらに加藤ら[4]は、模範解答との比較により、問題を抱える受講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ミング演習向け授業システムを提案している。これらは、受講生の理解状況を把握することを可能にするものであり、本研究の TA への質問のハードルを下げることは別のアプローチではあるが、組み合わせることにより授業をよりよくすることが可能になる。

栗原ら[5]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[6]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。本研究も、コミュニケーションがより難しくなるオンラインにおいて、受講生の TA への質問時や TA 自身が消極的な場合でも使えるようなシステムデザインを目指しており、一致する方向性である。

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高まると考えられる。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。

以上の、質問するハードルが高い問題を、まず TA に直接質問するのではなく、一旦システムに対して質問を投稿し、その質問を見た TA に呼んでもらうことで、「TA に呼ばれたから質問できる」と立場を変え、気軽に質問できるようにする。

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出来るようにし、TA 自身が回答できない内容であれば、その質問を別の TA に頼むことが出来るようにする。こうすることで、必ず質問が来たら対応しないとイケない TA の義務感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出来る。

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 サービスとして実装を行った。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うようにした。

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。

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れるようにした。ここで、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。

TA が質問に対応する際には、そのプログラムなどの把握が重要である。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、質問の取り下げ (図 4) を設定可能とした。なお、対応中の TA の名前や互いのマウスのカーソルも表示することで、質問対応を容易化している。また、不正行為防止のため、受講生は他者の質問の中身を見れないようにした。

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。ここで単純な通知では気づかない可能性があるため、5 秒程度の効果音も鳴らすことで気づきを促した。

本システムでは、当学科のプログラミング教育に 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、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。

4.2 使い方

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。

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)

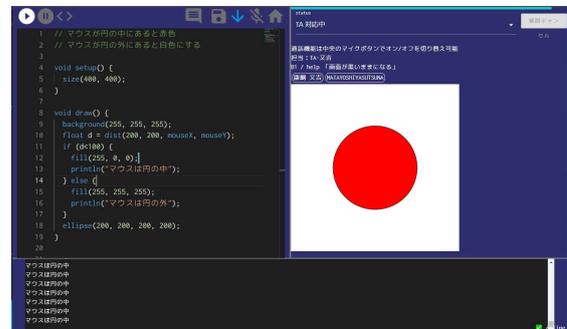


図 3. 受講生と TA がプログラムの共有や実行をし音声通話をしながら質問を解決するシェアエディタ



図 4. 受講生が質問を取り下げることが出来る質問キャンセルボタン

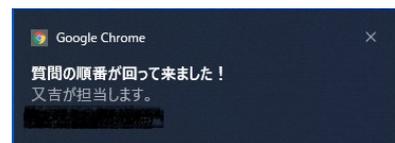


図 5. 受講生に順番が来たことを伝えるトースト通知

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の質問リストを把握することができる。また、質問をクリ

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、呼び出しがされる。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 運用形態

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 年生と再履修の学生による 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 を使用し、共著者が主として講義を実施していた(図 6)。

講義(100 分の 2 コマ)では、typing.run[1]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。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、簡易的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は askTA で質問してもらった。

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との関係のグラフを図 7 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 8 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 9 は、TA が「対応中」と「対応終了」にステ

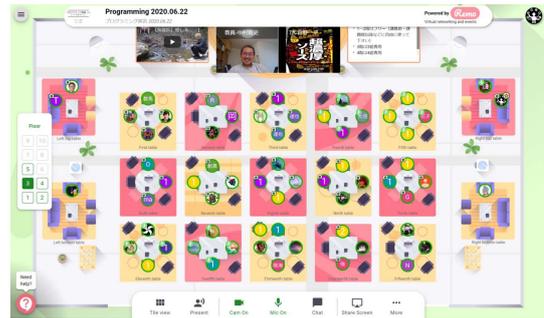


図 6. 中央のテーブルに受講生が座り、周りに TA が質問の待機している講義中の Remo Conference の様子

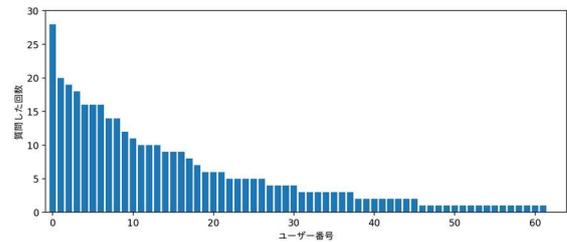


図 7. 質問した人とその回数の関係

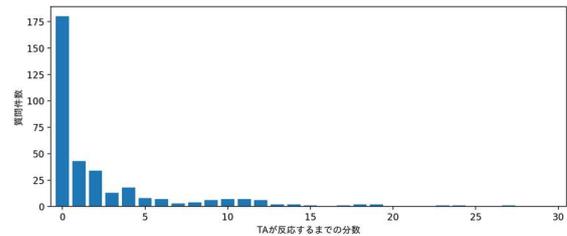


図 8. 受講生の質問に TA が反応するまでの待ち時間(分)の分布

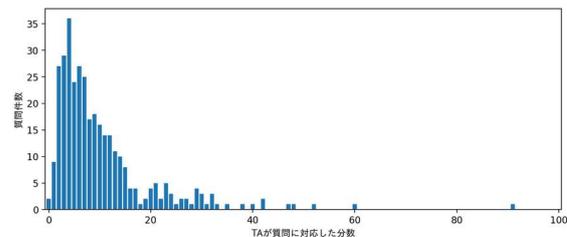


図 9. TA が受講生の質問に対応した時間(分)の分布

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)もシステムを運用しており、図 8 では待ち時間、図 9 では対応時間に休憩時間を含む質問があった。

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 の利用者は 61 名)から回答を得た。表 1 はそのアンケー

表 1. 受講生のアンケート項目とその分布と平均

質問項目	評価値の分布					平均
	-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 に質問する抵抗感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. TA のアンケート項目とその分布と平均

質問項目	評価値の分布					平均
	-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

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 1 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ことが分かる。

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」「ボタンを押し間違えた」「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。

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、著者を除く 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 2 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す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 に頼り切りな人がいた」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。

6 考察

6.1 利用状況の考察

図 1 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 2 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 3 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 の負担を下げることができた可能性がある。

askTA を運用した 8 回のうち 1 回の講義で、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。しかし、askTA により、特に大きな支障なく受講生は課題に取り組み、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。

6.2 受講生アンケートの考察

表 1 の結果より、受講生は今後も利用したいと感じており、また TA への抵抗を下げることでできていることも示唆された。

自由記述で得られたポジティブな意見は、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、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。実際に、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」などが一定数あった。これは、受講生に概要の書き方を指導しなかったことで起こったと考えられるため、指導により改善可能である。なお、質問の概要は講義実施者である教員が、課題解決に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。

6.3 TA アンケートの考察

表 2 の結果より、TA はシステムを好意的に捉えていたことが分かる。また、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、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 が対応できないことや、システムを使うことによって受講生に教えすぎてしまうことは、オンライン特有の柔軟性がない点やアクセスのしやすい点に起因していると考えられる。一方、TA も質問が来たことの通知が欲しいなどの機能に関する意見は、実装に反映し改善を行った。TA によってシステムの使い方や重視する機能が異なると考えられるため、意見を収集しながら機能や GUI 改善を行っていききたい。

7 まとめと今後

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。また、学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。

今後は、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、著者ら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[7]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。また、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。

謝辞

本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] 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. 情報処理学会 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, pp.1-8, 2020.
- [2] 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, pp.330-339, 2013.
- [3] 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, pp.2518-2527, 2013.
- [4] 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, pp.1819-1930, 2014.
- [5] 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。ピー・エヌ・エヌ新社, 2016.
- [6] 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, pp.45-53, 2010.
- [7] Y. Matayoshi, S. Nakamura. Abstract Thinking Description System for Programming Education Facilitation. In *Proc. HCII '20*, Vol. 12206, 2020.