

レスポンス・アナライザーの開発・分析に関する研究

海老原 宏樹¹ 近藤 郁哉¹ 高野 辰之² 小濱 隆司³ 宮川 治³

概要: 教員から学生へ演習課題を与え、学生がその課題を授業中に解く形態の授業では、課題につまずいている学生が授業に十分に参加できないという問題が存在する。その対策として、教員がクラス全体を見回りながら課題進捗状況を把握し、その状況に応じて授業の軌道修正をする必要があるが、クラス全体を見回すには労力や時間を要する。そこで、学生の反応状況をリアルタイムに分析できるレスポンス・アナライザーを活用する。本研究では教員から学生へ演習課題を与え、学生が授業中にその課題の解答ファイルをアップロードする形態の授業を想定し、ファイルアップロードシステムとレスポンス・アナライザーを組み合わせたシステムを開発する。本システムを使用することで、教員はクラス全体の課題進捗状況の把握が容易になり、学生は自身がアップロードしたファイルの評価結果をすぐに把握できる。その結果、学生の学習を促進させることが期待される。また、本システムを学生に使用させる実験を行い、そこで収集したファイルをもとに分析を行った。本稿では、本システムの詳細と実験の分析結果、本システムの使用感に関するアンケートの集計結果について報告する。

Study on development and analysis of response analyzer

1. はじめに

大学教育において、従来のような知識の伝達・注入を中心とした授業から、教員と学生が意思疎通を図りつつ、学生が主体的に問題を発見し解を見いだしていく能動的学修（アクティブ・ラーニング）への転換が必要とされている [1]。アクティブ・ラーニングでは、教員には学生の学習を促進する役割がある [2]。例えば、クラス全体の学習状況が良くない場合には授業の軌道修正をする必要がある。

アクティブ・ラーニングは多くの大学に様々な形態で導入されている。そのうちの一つに、教員が学生へ演習課題を与え、学生が授業中にその課題を解く形態がある [3][4]。その形態のアクティブ・ラーニングでは、学生が与えられた課題に手を付けられず、授業に十分に参加できないという問題が存在する。その問題の対策として、教員がクラス全体を見回りながら課題進捗状況を把握し、その状況に

じて授業の軌道修正をすることで、課題につまずいている学生を支援できると考えられる。しかし、クラス全体を見回すには労力や時間を要するため、教員の負担を軽減するためにはティーチング・アシスタント（TA）などによる支援が必要となる。

そこで、レスポンス・アナライザーを活用することでクラス全体の課題進捗状況の把握を容易にする。レスポンス・アナライザーは学生の反応状況をリアルタイムに分析できることから、授業進行の支援に有効であることが報告されている [5]。本研究では教員が学生へ演習課題を与え、学生が授業中に課題の解答ファイルをアップロードする形態の授業を想定する。想定する受講者数は 60 名程度である。本研究では教員と学生がアップロードできるファイルアップロードシステムと、教員と学生へリアルタイムにフィードバックを与えるレスポンス・アナライザーを組み合わせたシステムを開発する。本システムは授業中に学生からアップロードされたファイルの評価をリアルタイムに行い、その評価結果を教員と学生へフィードバックすることで学生の学習の促進を図る。

2. 関連研究

レスポンス・アナライザーを授業に導入し、学生の学習

¹ 東京電機大学大学院 情報環境学研究科
Graduate School of Information Environment, Tokyo Denki University

² 関東学院大学 理工学部
College of Science and Engineering, Kanto Gakuin University

³ 東京電機大学 情報環境学部
School of Information Environment, Tokyo Denki University

の促進を図る研究が行われている。

稲葉らは、授業中の学生の質問を促進し、教員が学生の質問や理解度をリアルタイムに把握できることを目的としたレスポンス・アナライザを開発した [6]。このシステムを授業に導入することで、学生の授業への参加および学習の動機づけの向上に有効であるとされている。

中島は、授業中の学生からのフィードバックと授業映像を統合し、授業後に視聴できるシステムを開発した [7]。このシステムは授業の改善に効果的であるとされている。

本研究は授業中に学生からアップロードされるファイルに着目している点でこれらの研究とは異なる。本システムを使用することで、ファイルをアップロードするのが遅い学生やアップロードしていない学生をリアルタイムに把握し、適切な対応を取ることができる。本研究の中で、本システムを使用した場合と使用しない場合について実用的規模で実験を実施し、本システムの効用を明らかにするとともに、その改善点を明確にする。

3. システム仕様

本システムは Web ブラウザ上で使用する。本システムのユーザは教員と学生で、両者ともファイルをアップロードできる。ここでは、本システムの仕様をファイルのアップロードと基本的な仕様の 2 点に分けて説明する。

3.1 ファイルのアップロード

本システムには教員と学生ともにファイルをアップロードできる。

3.1.1 教員がファイルをアップロードした場合

本システムは教員からアップロードされたファイルの形式に応じたコンテンツを生成する。

Java 言語のソースコードが記述されたファイルが教員からアップロードされた場合、そのファイルが模範解答となる課題が生成される。課題が生成されると、学生はその課題の解答ファイルをアップロードできるようになる。

JSON 形式のファイルが教員からアップロードされた場合、そのファイルのスキーマに応じたコンテンツが生成される。コンテンツの例として、学生に CSV 形式の解答ファイルをアップロードさせる課題があげられる。

その他の任意の形式のファイルが教員からアップロードされた場合、そのファイルを学生がダウンロードできるようになる。

3.1.2 学生がファイルをアップロードした場合

本システムは学生からアップロードされたファイルに対して、そのファイルの形式に応じた評価をリアルタイムに行う。

Java 言語のソースコードが記述されたファイルが学生からアップロードされた場合、コンパイル、インデント、枠組みの導出の 3 点を評価する。なお、これらの評価は長谷

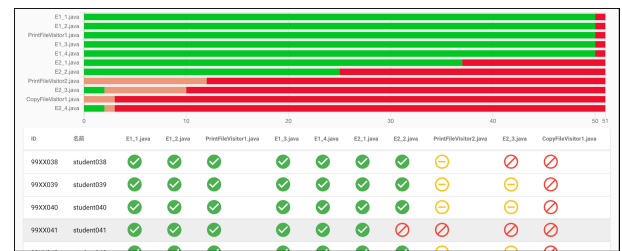


図 1 教員側の授業画面

Fig. 1 Lecture screen of teacher side

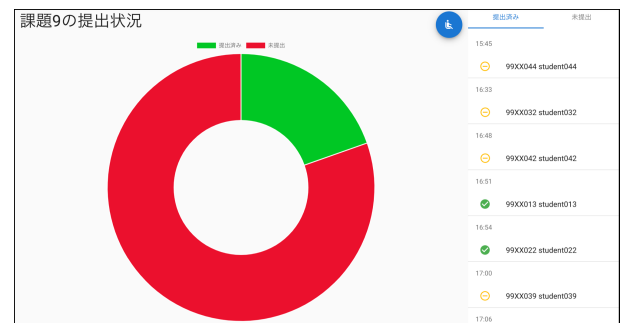


図 2 課題の進捗状況を表示した教員側の授業画面

Fig. 2 Lecture screen of teacher side showing the progress of a task

川らが開発したシステム [8] の評価機構を用いて行う。ただし、ユニットテスト用のテストケースが記述されたファイルが教員からアップロードされている場合は、実装の評価も行う。

CSV 形式のファイルが学生からアップロードされた場合、そのファイルに記述されたデータの構造が正しいかを評価する。評価する項目はヘッダ名、ヘッダ・バリューの順番、ヘッダ・バリューの数、バリューの型の 4 点である。

3.2 基本的な仕様

ここでは、教員側と学生側それぞれの授業画面とその仕様について説明する。

3.2.1 教員側の仕様

教員側の授業画面ではクラス全体の課題進捗状況を把握できる。教員側の授業画面を図 1 に示す。画面上部のグラフはクラス全体の課題進捗状況を色で表している。画面下部の表は学生一人ひとりの課題の評価結果をアイコンで表している。

教員側の授業画面では各課題の細かい進捗状況も把握できる。各課題の進捗状況を表示した授業画面を図 2 に示す。この画面では教員が選択した課題の進捗状況が円グラフで表示される。画面右側のリストでは課題を提出済みの学生、未提出の学生をそれぞれ確認できる。

また、教員は学生からアップロードされたファイルの評価結果詳細と内容を確認できる。ファイルの評価結果詳細と内容が表示された状態の教員側の授業画面を図 3 に示す。



図 3 ファイルの評価結果詳細と内容を表示した教員側の授業画面
Fig. 3 Lecture screen of teacher side showing the evaluation details and contents of a file



図 5 ファイルの評価結果詳細を表示した学生側の授業画面
Fig. 5 Lecture screen of student side showing the evaluation details of a file



図 4 学生側の授業画面
Fig. 4 Lecture screen of student side

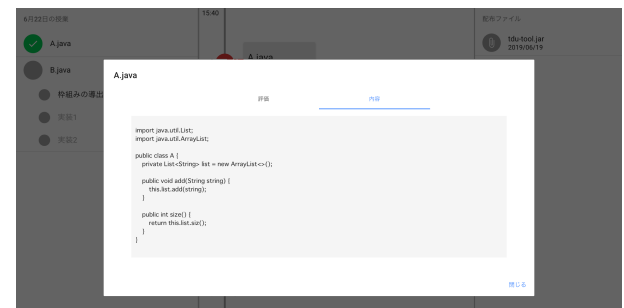


図 6 ファイルの内容を表示した学生側の授業画面
Fig. 6 Lecture screen of student side showing contents of a file

3.2.2 学生側の仕様

学生側の授業画面は課題一覧、タイムライン、配布ファイル一覧で構成されている。学生はこの画面上へファイルをドラッグアンドドロップすることでアップロードできる。アップロードされたファイルは即時に評価され、その評価結果がこの画面上へ反映される。学生側の授業画面を図 4 に示す。課題一覧では、授業で行う課題とその評価結果を表すアイコンが表示される。タイムラインでは、学生自身がアップロードしたファイルとその評価結果を表すアイコンが時系列で表示される。また、課題一覧とタイムラインに表示されているアイコンをクリックすることで、そのファイルの評価結果詳細とファイルの内容を確認できる。ファイルの評価結果詳細が表示された状態の授業画面を図 5、ファイルの内容が表示された状態の授業画面を図 6 に示す。配布ファイル一覧では、教員から配布されているファイルの一覧が表示されており、ファイルをクリックすることでそのファイルをダウンロードできる。

4. 実装

本システムはクライアントとサーバによって構成されている。クライアントとサーバ間は HTTP を用いて通信を行っている。

サーバではクライアントからの HTTP リクエストを受け取るための REST API を実装している。本システムの REST API はマイクロサービスアーキテクチャ [9] に基づいて設計している。ここでは、マイクロサービスアーキテクチャの利点とマイクロサービスアーキテクチャを適用し

た本システムの REST API について説明する。

4.1 マイクロサービスアーキテクチャ

分割されていない一つのモジュールで構成されているモノリシックサービスの場合、新しい機能が追加されるとそのサービスの規模も大きくなる。大規模なモノリシックサービスになると、変更が必要な箇所の特定が困難になる。そういった大規模なモノリシックサービスの問題点を解決できるのがマイクロサービスである。マイクロサービスアーキテクチャは、大規模な一つのモノリシックサービスではなく、小規模で独立した複数のマイクロサービスで構成するアーキテクチャである。マイクロサービスアーキテクチャの利点として次のようなものが挙げられる。

4.1.1 技術異質性

マイクロサービスアーキテクチャでは複数のサービスがそれぞれ独立しているため、サービスごとに異なる技術を使用できる。モノリシックサービスの場合、使用している技術を変更してしまうとその変更がシステム全体に大きな影響を及ぼす可能性がある。そのため、使用する技術を決するまでに多くの時間を要する。それに対してマイクロサービスの場合、そのサービスで使用している技術を変更しても他のサービスへの影響は少ない。そのため、新しい技術も採用しやすく、使用する技術を迅速に決定できる。

4.1.2 スケーリング

マイクロサービスアーキテクチャでは複数のサービスがそれぞれ独立しているため、スケーリングが必要なサービ

スだけを対象にスケールアップができる。モノリシックサービスでスケールアップが必要な場合、その大規模なサービスの一つとしてスケールアップしなければならない。それに対してマイクロサービスの場合、スケールアップが必要なサービスのみをスケールアップできるため、スケールアップが不要な他のサービスには影響しない。

4.1.3 デプロイの容易性

マイクロサービスアーキテクチャでは複数のサービスがそれぞれ独立しているため、一つのサービスを他のサービスとは独立してデプロイできる。モノリシックサービスの一部を変更し、その変更をリリースする場合、大規模なサービス全体をデプロイしなくてはならない。このようなデプロイは影響範囲が大きくリスクが高くなる可能性がある。それに対してマイクロサービスの場合、一つのサービスに変更を加えても他のサービスとは独立してデプロイができる。デプロイによって問題が生じた場合は、問題の原因であるサービスを迅速に特定できる。

4.2 各サービスの説明

本システムの REST API は 6 個のサービスで構成されている。ここでは、それぞれのサービスがどのような役割を担っているのかを説明する。本システムの REST API の構成の概略を図 7 に示す。

4.2.1 BFF

BFF は Backends For Frontends の略称である [10]。マイクロサービスアーキテクチャを適用する場合、サービスが複数個に分割されてしまうため、クライアントがそれぞれのサービスを呼び出す必要がある。また、いずれかのサービスに変更が生じた場合、クライアントにもその変更が影響する可能性もある。そこで、クライアントが複数のサービスを使用しやすくするために、複数のサービスの呼び出しをまとめたサービスである BFF を導入した。BFF を導入することでクライアントは BFF のみを呼び出すことになるため、複数のサービスを呼び出す必要がなくなる。

4.2.2 科目・授業・課題

このサービスは本システムで扱う科目、授業、課題の管理をしている。教員や学生が科目等にアクセスするための権限の管理もしている。

4.2.3 ファイル提出

学生からファイルがアップロードされた場合、そのファイルは BFF からこのサービスへと渡される。学生からアップロードされたファイルの形式に応じたファイル評価サービスを呼び出し、そのサービスから返ってくる評価結果をデータベースへと格納するのがこのサービスの役割である。また、教員が事前にアップロードする模範解答用のファイルもこのサービスが管理している。

4.2.4 Java ファイル評価

このサービスはアップロードされた Java 形式のファイル

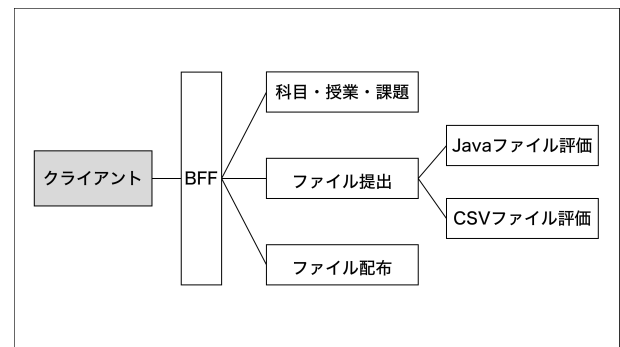


図 7 本システムの REST API の構成

Fig. 7 Configuration of REST API of this system

の評価を行う。評価対象のファイル、評価対象以外のファイル、模範解答用のファイルをこのサービスへ送ることで評価結果が返ってくる。それぞれのファイルは Java 言語のソースコードが記述された Java 形式のファイルである。評価項目はコンパイル、インデント、枠組みの導出、実装の 4 点である。ただし、実装を評価する場合は、模範解答用のファイル以外にユニットテスト用のテストケースが記述されたファイルが必要となる。

4.2.5 CSV ファイル評価

このサービスはアップロードされた CSV 形式のファイルに記述されたデータの構造が正しいかを評価する。評価対象の CSV 形式のファイル、模範解答用の JSON 形式のファイルをこのサービスへ送ることで評価結果が返ってくる。評価項目はヘッダ名、ヘッダ・バリューの順番、ヘッダ・バリューの数、バリューの型の 4 点である。CSV 形式のファイルの評価は、アップロードされた CSV 形式のファイルを XML 形式に変換し、そのファイルを XML Schema を用いて評価を行っている。XML Schema は XML ファイルの構造を定義するためのスキーマ言語である。

4.2.6 ファイル配布

このサービスは教員から学生に配布するファイルの管理をしている。学生がファイルをダウンロードした時、そのファイルのダウンロードのログを記録している。

5. 実験 1

本研究で想定している規模の授業で本システムが動作するかを確認するために、本学のシステムデザイン工学部で開講されている情報システム工学実験 I で本システムを使用し、実験を行った。ここでは、実験の結果と実験で収集したファイルをもとに行った分析の結果について説明する。

5.1 実験 1 の実験概要

実験 1 では実験を 3 回行い、それぞれの実験は指定された仕様を満たすプログラムを Java 言語を用いて被験者に作成させ、作成した Java 形式のファイルを本システムにアップロードする演習形態で行った。それぞれの実験の被験者

数は 50 名程度で、3 回分の実験の被験者数の合計は 147 名である。被験者に提示した課題の数は 11 個である。本システムの動作環境は CPU のクロック周波数が 2.10GHz、CPU のコア数が 2 個、メモリが 8GB である。なお、今回の実験ではコンパイル、インデント、枠組みの導出の 3 点を評価した。

5.2 実験 1 の実験結果

実験 1 の結果、50 名程度の規模の授業で本システムを使用しても基本的な動作に問題はなかった。しかし、現時点の本システムではクラスの枠組みの導出の評価を厳密に行うため、学生がアップロードしたファイルに不必要なインスタンス変数やメソッドが定義されている場合は不正解となる。そのため、学生にプログラムを自由に実装させ、そのプログラムが指定された仕様通りであるかを評価することが困難となる。より実用的なシステムにするためには、枠組みの導出の評価の有無を柔軟に切り替えられるような機能の追加が必要であると考えられる。

5.3 実験 1 の分析結果

実験 1 で提示した 11 個の課題のうち、被験者全員が時間内に終了した 5 個の課題を分析の対象とした。そこで、ファイルをアップロードした回数（以下 試行回数と呼ぶ）と各被験者の 5 個の課題を終了するまでに要した時間（以下 終了時間と呼ぶ）を調べ、その相関関係を求めた。終了時間は、「被験者が最初にアップロードしたファイルの更新時間」と「対象の 5 個の課題を全て正解にした時点で被験者が最後にアップロードしたファイルの更新時間」の差で求めた。散布図を図 8、分析結果の詳細を表 1 に示す。散布図は縦軸が試行回数、横軸が終了時間である。また、試行回数が多く終了時間が早い被験者のグループ（図 8 ①）、試行回数が少なく終了時間が遅い被験者のグループ（図 8 ②と③）、試行回数が多く終了時間が遅い被験者のグループ（図 8 ④）の 3 つのグループに分けてそれぞれのグループの分析を行った。グループ分けは、試行回数と終了時間それぞれの平均と標準偏差を足した値を基準に行った。

5.3.1 試行回数が多く終了時間が早い被験者のグループ

このグループに該当した被験者数は 9 名であった。このグループでは主に 3 つの傾向が見られた。1 つ目は被験者自身の PC 上でコンパイルをしない傾向である。コンパイルエラーのあるファイルをアップロードした場合、ファイルの評価結果は不正解になる。そのため、コンパイルをせずにアップロードしたファイルが不正解となり、試行回数が多くなったと考えられる。2 つ目はソースコードのインデントのミスが多い傾向である。本システムのフィードバックを確認しながらソースコードのインデントを修正するという過程を繰り返したため、試行回数が多くなったと考えられる。3 つ目はシステムの使い方に苦戦してい

表 1 分析結果の詳細（実験 1）

Table 1 Details of analysis result (experiment1)

	平均	標準偏差
試行回数 (回)	7.76	4.56
終了時間 (秒)	3125.52	1563.21

た傾向である。本システムでは、コンパイルエラーのあるファイルがアップロードされた状態で他のファイルをアップロードした場合はそのファイルの評価結果が不正解になる。また、依存関係のあるファイルをアップロードする場合は、正しい順番でファイルをアップロードしなければ不正解になってしまう場合がある。こういったシステムの仕様を理解するために試行を重ねたため、試行回数が多くなったと考えられる。

5.3.2 試行回数が少なく終了時間が遅い被験者のグループ

このグループに該当した被験者数は 15 名であった。このグループでは主に 2 つの傾向が見られた。1 つ目は被験者自身の PC 上でファイルを複数個作り、それらのファイルを連続でアップロードする傾向である。ファイルを 1 つ作成したらシステムにアップロードし、フィードバックを確認して修正を行うというシステムの使い方をすることで終了時間が短縮されるのではないかと考えられる。2 つ目は対象の 5 個の課題を全て正解にする前に、それ以降の課題に着手する傾向である。課題を 1 つ正解にしたら次の課題へ進むというシステムの使い方をすることで終了時間が短縮されるのではないかと考えられる。

また、このグループの中でも特に終了時間の遅い被験者 5 名（図 8 ③）について確認したところ、4 名の被験者が 5 個の課題を全て正解にする前にそれ以降の課題に着手していた。残りの 1 名の解答を確認したところ、先の課題には着手していなかったが、試行回数は 6 回だった。

5.3.3 試行回数が多く終了時間が遅い被験者のグループ

このグループに該当した被験者数は 4 名であった。このグループでは主に 2 つの傾向が見られた。1 つ目はシステムのフィードバックに気づくのが遅い、あるいは見ていない傾向である。本システムの UI がわかりづらかったため、システムのフィードバックに気づくのが困難であったと考えられる。なお、より使いやすいシステムにするために、本実験終了後に本システムの UI を修正した。2 つ目はソースコードのインデントのミスが多い傾向である。本システムのフィードバックを確認しながらソースコードのインデントを修正するという過程を繰り返したため、試行回数が多く、終了時間が遅くなったと考えられる。

5.4 実験 1 の考察

被験者がアップロードしたファイルが本システムによって即時に評価され、被験者はそのフィードバックによってファイルの修正をすることができた。そのため、ソース

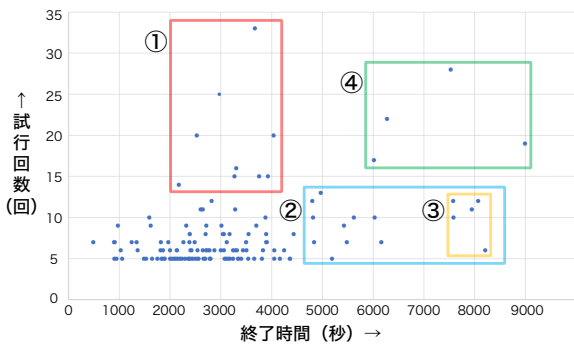


図 8 分析結果の散布図 (実験 1)

Fig. 8 Scatter plot of analysis result (experiment1)

コードのインデントといった教員や TA が直接指摘しなければ気づけないようなミスに被験者自身が気づくことができた。本システムがない場合は教員や TA が指摘するまでに時間を要したり、修正箇所を見落とす場合があるが、本システムによってそれらの問題を軽減できると考えられる。また、システムの使い方に苦戦していた被験者が多く見られた点については、自身の PC 上でファイルをコンパイルし、コンパイルエラーのないファイルを正しい順番でアップロードすることでシステムを容易に使用できるようになると考えられる。

6. 実験 2

本システムが学生に対してどのような効果を与えているのかを確認するために、本学のシステムデザイン工学部で開講されているコンピュータ・プログラミング III で本システムを使用し、実験を行った。ここでは、実験で収集したファイルをもとに行った分析の結果と、本システムの使用感に関するアンケートの集計結果について説明する。

6.1 実験 2 の実験概要

実験 2 は指定された仕様を満たすプログラムを Java 言語を用いて被験者に作成させ、作成した Java 形式のファイルを本システムにアップロードする演習形態で行った。また、実験 2 は本システムを使用しながら演習を行うグループ (グループ A) と本システムを使用せずに演習を行うグループ (グループ B) に分けて行った。なお、グループ B の被験者には演習終了後に本システムに解答ファイルをアップロードさせた。

演習の時間は 60 分で、被験者数はグループ A が 43 名、グループ B が 45 名である。被験者は Java 言語を用いた手続き型プログラミングを学習する科目を履修済みで、オブジェクト指向プログラミングを学習中の初学者である。被験者全員が同じ科目を履修済みであることから、被験者の習熟度に大きな差はないと考えられる。なお、今回の実験ではコンパイル、インデント、クラスの枠組みの導出、実装の 4 点を評価した。被験者に提示した課題は次の通りで

ある。

課題 1 インスタンス変数、コンストラクタ、インスタンス

変数を返却するメソッドを持つクラスの枠組みの導出

課題 2 インスタンス変数、コンストラクタ、インスタンス

変数を返却するメソッドを持つクラスの実装

課題 3 インスタンス変数、コンストラクタ、ArrayList に

処理を委譲するメソッド、ArrayList と for 文を用いて

処理をするメソッドを持ち、課題 2 で作成したクラス

に依存したクラスの実装

6.2 実験 2 の分析結果

実験 2 で収集したファイルをもとに課題ごとに分析を行った。ここでは、グループ A とグループ B の分析結果を比較する。

6.2.1 課題 1 の分析結果

各グループの課題 1 の評価結果を表 2 に示す。課題 1 の総合評価が正解の被験者はグループ A の方が多かった。グループ A ではコンパイルとインデントの評価が不正解の被験者が 0 名だが、グループ B ではコンパイルが不正解の被験者が 4 名、インデントが不正解の被験者が 3 名であった。これは、グループ A の被験者は本システムのフィードバックを確認しながら演習を行ったためであると考えられる。また、課題 1 は 3 個の課題の中で最も簡単な課題だが、グループ B では不正解の被験者が 13 名いた。不正解の被験者のソースコードを確認したところ、課題 1 はクラスの枠組みの導出までを行う課題だが、クラスの実装まで進めてしまった被験者が多く見られた。本システムを使用することで、このような勘違いをしてしまう被験者を減らすことができると考えられる。

6.2.2 課題 2 の分析結果

各グループの課題 2 の評価結果を表 3 に示す。総合評価が正解の被験者はグループ A の方が多かった。グループ A ではコンパイルとインデントの評価が不正解の被験者が 0 名だが、グループ B ではコンパイルが不正解の被験者が 4 名、インデントが不正解の被験者が 1 名であった。これも課題 1 と同様の理由であると考えられる。

6.2.3 課題 3 の分析結果

各グループの課題 3 の評価結果を表 4 に示す。総合評価が正解の被験者はグループ A の方が多かった。しかし、グループ A では課題 3 を提出していない被験者が 12 名いた。これは、グループ A の被験者は本システムのフィードバックを確認しながら課題を正解にしていくため、課題 3 に着手できずに演習が終了してしまったためであると考えられる。このことから、本システムを使用することで課題の解答の精度は高くなるが、課題を終了するまでに要する時間も増える可能性があることがわかる。

6.2.4 正解した課題数

各グループの正解した課題数の詳細を表 5 に示す。グ

表 2 課題 1 の評価結果 (実験 2)

Table 2 Evaluation result of task1 (experiment2)

評価項目	グループ A			グループ B		
	正	不	未	正	不	未
コンパイル	43	0	-	41	4	-
インデント	43	0	-	42	3	-
枠組みの導出	42	1	-	37	8	-
実装	40	3	-	32	13	-
総合評価	40	3	0	32	13	0

正: 正解 不: 不正解 未: 未提出

表 3 課題 2 の評価結果 (実験 2)

Table 3 Evaluation result of task2 (experiment2)

評価項目	グループ A			グループ B		
	正	不	未	正	不	未
コンパイル	43	0	-	41	4	-
インデント	43	0	-	44	1	-
枠組みの導出	42	1	-	38	7	-
実装	40	3	-	38	7	-
総合評価	40	3	0	38	7	0

正: 正解 不: 不正解 未: 未提出

表 4 課題 3 の評価結果 (実験 2)

Table 4 Evaluation result of task3 (experiment2)

評価項目	グループ A			グループ B		
	正	不	未	正	不	未
コンパイル	25	6	-	34	10	-
インデント	31	0	-	42	2	-
枠組みの導出	28	3	-	30	14	-
実装	17	14	-	12	32	-
総合評価	17	14	12	11	33	1

正: 正解 不: 不正解 未: 未提出

表 5 正解した課題の数 (実験 2)

Table 5 Number of correct tasks (experiment2)

正解数	グループ A		グループ B	
	人数	割合 (%)	人数	割合 (%)
3 個	17	39.5	9	20.0
2 個	22	51.2	23	51.1
1 個	2	4.7	8	17.8
0 個	2	4.7	5	11.1

グループ A では課題を 2 個以上正解した被験者が 9 割程度であるのに対して、グループ B では 7 割程度であった。このことから、本システムを使用することで課題の解答の精度が高くなることが考えられる。また、グループ B では 3 個の課題のうち最も難しい課題である課題 3 が正解であるのに、それよりも簡単な課題 1 や課題 2 が不正解である被験者が数名いた。グループ A ではこのような被験者は見られなかったため、本システムを使用することでこのような被験者を減らすことができると考えられる。

6.3 実験 2 のアンケート結果

実験 2 の演習終了後に本システムの使用感に関するアンケートを行った。アンケートの質問項目は次の通りである。

質問 1 本システムの UI は使いやすかったか

質問 2 本システムのフィードバックはわかりやすかったか

質問 3 本システムのフィードバックはコーディングの手助けになったか

また、3 個の質問以外に自由記述の回答欄を設けた。アンケートに回答した被験者数はグループ A が 34 名、グループ B が 40 名である。なお、グループ A の被験者には本システムの使い方を詳しく説明したが、グループ B の被験者には本システムにファイルをアップロードする方法のみを説明した。ここでは、グループ A とグループ B のアンケートの集計結果を比較する。

6.3.1 質問 1 の集計結果

各グループの質問 1 の集計結果を表 6 に示す。グループ A では満足とやや満足に回答した被験者が 7 割以上おり、不満とやや不満に回答した被験者は 0 名だった。また、グループ B でも満足とやや満足に回答した被験者が 6 割弱いたことから、本システムの UI は使いやすいのではないかと考えられる。

6.3.2 質問 2 の集計結果

各グループの質問 2 の集計結果を表 7 に示す。グループ A とグループ B ともに、満足とやや満足に回答した被験者が 5 割以上いた。不満とやや不満に回答した被験者が少数いることについては、自由記述の回答を確認したところ、不正解時のフィードバックのメッセージをより詳しくしてほしいという意見がグループ B で 1 件あった。

6.3.3 質問 3 の集計結果

各グループの質問 3 の集計結果を表 8 に示す。グループ A では満足とやや満足に回答した被験者が 4 割以上いたが、不満とやや不満に回答した被験者が 2 割弱いた。グループ A で不満とやや不満に回答した被験者の解答ファイルを確認したところ、課題 3 が未提出の被験者が多く見られた。本システムを使用しても 3 個の課題を終了できなかったため、本システムのフィードバックがコーディングの手助けにはならないと感じたのではないかと考えられる。

6.3.4 自由記述の回答

グループ A の回答として、「演習だけでなく授業中にも導入できれば自主的に授業中の課題をこなせると思った」、「ファイルをアップロードしたらすぐ評価してくれるのが良いと思った」、「自分がいつ提出したのかが明確になるので、どの問題にどれだけ時間を要したのかがわかりやすかった」、「課題が不正解の場合、どこが間違っているのかを詳しく教えてほしい」といった回答があった。グループ A の被験者は本システムを使用しながら演習を行ったため、普段の授業でも使えるのではないかと意見や、課

表 6 質問 1 の集計結果 (実験 2)

Table 6 Aggregate result of question1 (experiment2)

回答	グループ A		グループ B	
	人数	割合 (%)	人数	割合 (%)
満足	21	61.8	11	27.5
やや満足	5	14.7	12	30.0
普通	8	23.5	14	35.0
やや不満	0	0.0	3	7.5
不満	0	0.0	0	0.0

表 7 質問 2 の集計結果 (実験 2)

Table 7 Aggregate result of question2 (experiment2)

回答	グループ A		グループ B	
	人数	割合 (%)	人数	割合 (%)
満足	14	41.2	11	27.5
やや満足	4	11.8	12	30.0
普通	13	38.2	13	32.5
やや不満	2	5.9	4	10.0
不満	1	2.9	0	0.0

表 8 質問 3 の集計結果 (実験 2)

Table 8 Aggregate result of question3 (experiment2)

回答	グループ A		グループ B	
	人数	割合 (%)	人数	割合 (%)
満足	12	35.3	10	25.0
やや満足	4	11.8	10	25.0
普通	12	35.3	17	42.5
やや不満	5	14.7	2	5.0
不満	1	2.9	1	2.5

題を提出した時間に関する意見が見られた。

グループ B の回答として、「どこを間違えたのかすぐにわかるのはとても助かる」、「提出した時点で課題が正解かわかるのは助かる」、「実装できていたものが不正解になり、フィードバックのメッセージを見たが何が何だかわからなかった」、「課題の順番通りに提出していかないと不正解になってしまうのをどうにかしてほしいと感じた」といった回答があった。グループ B では演習終了後に課題を提出させたため、課題の順番通りに提出しないと不正解になってしまうという意見が見られた。

6.4 実験 2 の考察

本システムを使用した被験者の方が課題の正解数が多かったことから、課題の解答の精度が高くなることがわかった。しかし、本システムを使用した被験者の方が課題未提出者数が多かったことから、本システムを使用すると課題に要する時間が増えることがわかった。そのため、本システムは学生に課題を正確に行わせる目的には効果を発揮するが、課題を早く行わせる目的には効果を発揮しないと考えられる。

また、本システムを使用することで課題が不正解の学生や未提出の学生を画面上でリアルタイムに把握できる。そのため、課題につまずいている学生の早期発見に繋がりが、学生へのサポートが必要なおきに対応できるようになると考えられる。

7. まとめ

本研究では学生の学習を促進させるために、教員と学生がファイルをアップロードできるファイルアップロードシステムと、教員と学生へリアルタイムにフィードバックを与えるレスポンス・アナライザを組み合わせたシステムを開発した。教員が本システムにファイルをアップロードした場合、そのファイルの形式に応じたコンテンツを生成する。学生が本システムにファイルをアップロードした場合、そのファイルをリアルタイムに評価し、その評価結果を教員と学生へフィードバックする。学生からアップロードされたファイルの評価結果をリアルタイムにフィードバックすることで、教員はクラス全体の課題進捗状況の把握が容易になり、学生は自身がアップロードしたファイルの評価結果が把握できる。本システムを使用することで授業の軌道修正に要する時間が短縮され、学生の学習をより促進させることが期待される。

参考文献

- [1] 文部科学省:新たな未来を築くための大学教育の質的転換に向けて (online), <http://www.mext.go.jp/component/b_menu/shingi/toushin/_icsFiles/afieldfile/2012/10/04/1325048_1.pdf> (2018.05.30), p.9.
- [2] 中井俊樹:アクティブラーニング (シリーズ大学の教授法), 玉川大学出版部 (2015), p.15.
- [3] 溝上慎一:アクティブ・ラーニング導入の実践的課題, 名古屋高等教育研究 第 7 号 (2007), pp.273-280.
- [4] 玉川大学:アクティブ・ラーニング事例集 (online), <<http://www.tamagawa.jp/university/introduction/outline/u-ap/pdf/index-pdf-01.pdf>> (2018.06.20), pp.42-46.
- [5] 永岡慶三:レスポンス・アナライザを用いた授業進行支援システムの開発, 日本教育工学雑誌 10 巻 3 号 (1986), pp.11-18.
- [6] 稲葉利江子, 山肩洋子, 大山牧子, 村上正行:発言の自由度を高めたレスポンスアナライザを活用した大学授業の実践と評価, 日本教育工学会論文誌 36 巻 3 号 (2012), pp.271-279.
- [7] 中島平:レスポンスアナライザによるリアルタイムフィードバックと授業映像の統合による授業改善の支援, 日本教育工学会論文誌 32 巻 2 号 (2008), pp.169-179.
- [8] 長谷川伸, 松田承一, 高野辰之, 宮川治:プログラミング入門教育を対象としたリアルタイム授業支援システム, 情報処理学会論文誌 52 巻 12 号 (2011), pp.3135-3149.
- [9] Sam Newman 著, 佐藤直生監訳, 木下哲也訳:マイクロサービスアーキテクチャ, オライリー・ジャパン (2016), pp.1-13.
- [10] Sam Newman 著, 佐藤直生監訳, 木下哲也訳:マイクロサービスアーキテクチャ, オライリー・ジャパン (2016), pp.83-85.