

2024年9月14日



日本ムードル協会全国大会（2024） 発表論文集

PROCEEDINGS OF MOODLEMOT JAPAN 2024 ANNUAL CONFERENCE

日本ムードル協会 / MOODLE ASSOCIATION OF JAPAN

目次^a

査読付き論文

Moodle での e-Learning 環境の機能改善の報告	6
亀田真澄, 宇田川暢	
大学初修生物学におけるオンラインテスト実施方法の変更による成績評価改善の検証	15
中馬いづみ, 得字圭彦, 斉藤準	
Moodle に埋め込まれた動的幾何コンテンツに対する JavaScript を用いた学習分析に向けて	21
斉藤準	
フロントエンド API による Moodle と Jupyter 環境の連携	28
斉藤準	

査読なし論文

学生の省察支援ツールとしての Moodle 活用～活動モジュール比較検討～	38
浅田義和, 村岡千種	

INDEX^b

Refereed papers

Report on improvements to the functionality of the e-Learning environment in Moodle	6
MASUMI KAMEDA, MITSURU UDAGAWA	
Verification of the improvement in grading methods through modified online test administration for an undergraduate introductory Biology course	15
IZUMI CHUMA, YOSHIHIKO TOKUJI, JUN SAITO	
Using JavaScript in Moodle Standard Modules	21
JUN SAITO	
Connecting Moodle with the Jupyter Environment Through a Frontend API	28
JUN SAITO	

Non-refereed papers

Moodle Usage as a Tool for Reflection Support- Comparative review of activity modules -	38
YOSHIKAZU ASADA, CHIKUSA MURAOKA	

序文

日本ムードル協会主催の MoodleMoot Japan 2024 は、2024 年 2 月 16 日(金)~2 月 18 日(日)に長崎国際大学にて開催されました。この論文集が発表された内容のさらなる共有につながるだけでなく、参加が叶わなかった多くの方にとって活用されることを期待します。

この研究論文集には、国会図書館発行の ISSN (International Standard Serial Number) が付与され、この論文集の表ページの一番下に記されています。

今回も、論文に査読付き部門と査読なし部門を設けましたので、掲載論文が各自の研究履歴として残るようになりました。今回は、4 本の査読付き論文と 1 本の査読なし論文を掲載します。発表者の皆様におかれましては、積極的な投稿をいただけますことを期待しております。

最後に、査読者の先生方に心から感謝いたします。

Preface

MoodleMoot Japan 2024, hosted by the Moodle Association of Japan, was held from Friday, February 16, 2024 to Sunday, February 18, 2024. The conference was held at Nagasaki International University. We hope that this collection of papers will not only lead to further sharing of the published content but will also be of use to many people who were unable to participate.

The ISSN (International Standard Serial Number), which is shown at the bottom of the top page, is assigned by the Japan National Diet Library so that the papers published can be cited by the ISSN and people can utilize their publications as their research career.

This year we had three refereed papers and one non-refereed papers. I hope the participants of Moot will utilize and contribute to the proceedings more.

Finally, I would like to share my appreciation for the referees' effort.

編集者 八木(佐伯) 街子 / Editor Machiko Saeki Yagi

日本ムードル協会全国大会発表論文集 12 巻

2024 年 9 月 14 日発行

発行人：浅田 義和

発行所：自治医科大学医学教育センター

〒329-0498 栃木県下野市薬師寺 3311-1

ISSN 2189-5139

査読付き論文 / Refereed Papers

Moodle での e-Learning 環境の機能改善の報告

亀田 真澄^{1,a)} 宇田川 暢^{2,b)}

概要：2012年から筆者らは、旧来型対面授業（対象：理工系大学）との融合を目指し、履修者が教室外で自律的に学習できるように、e-Learning (Moodle) サイトを構築し、授業のデジタル化を進めている。筆者らはこの教育研究活動を、大学教育における「デジタイゼーション (Digitization)」の一面として捉えている。2020年の「新型コロナ禍 (COVID-19)」以降、大学教育では「非接触化」と「遠隔化」されたニューノーマル時代の授業変革が必要とされた。筆者らはこの変革を大学教育における「デジタイゼーション (Digitalization)」の一面として捉えている。これらのさらなる高度化が教育における「デジタルトランスフォーメーション (Digital Transformation)」すなわち「教育 DX」や「EduTech」である。2019年に筆者らは日本ムードル協会 (Moodle Association of Japan、略称「MAJ」と記す。)の補助金助成制度プログラムに「S-P 表プラグイン開発と実践」という研究テーマで応募し、採択された。その結果、大学教育におけるデジタイゼーションに関する研究活動を開始する運びになった。続けて、「Moodle Moot Japan 2020」大会で『「S-P 曲線」を生成するプラグイン」というテーマで初期の研究成果を発表した。その後の継続的な研究開発により、最近では「注意係数 (学習者) × 評点」の散布図を生成できるようになった。本稿の前半部分では、この改良された「S-P 表分析」プラグインについて紹介する。筆者らが運用している 数学教育向け e-Learning システムにおいて、「STACK」と「Maxima」が主要な機能である。本稿の後半部分では、STACK のパフォーマンスに焦点を当てた研究活動について報告する。筆者らは、高等教育における数学教育の質の向上を目指し、EdTech に関するこれらの研究活動に継続して取り組んでいる。

Report on improvements to the functionality of the e-Learning environment in Moodle

MASUMI KAMEDA^{1,a)} MITSURU UDAGAWA^{2,b)}

Abstract: Since 2012, the authors have been integrating traditional face-to-face STEM education at universities with e-Learning platforms such as Moodle, aiming to empower students to learn autonomously outside the classroom. This effort is part of their broader strategy to digitize university education. The onset of the COVID-19 pandemic in 2020 accelerated the shift towards "new normal" educational practices, emphasizing remote and contactless learning in higher education. This highlighted the critical role of digital transformation ("EduTech") in adapting to new teaching paradigms. In response to these challenges, the authors have continued their research and development in EduTech, with a specific focus on enhancing mathematical education through innovative tools embedded within their e-Learning system. Key functionalities like "STACK" and "Maxima" play pivotal roles in this system, facilitating interactive and personalized learning experiences for students. Their work has been supported by grants from the Moodle Association of Japan, enabling them to develop plugins and present their findings at prestigious conferences such as Moodle Moot Japan 2020. Their ongoing research has evolved to include sophisticated features such as generating scatter plots to analyze attention coefficients and grades ("S-P table analysis"). These advancements underscore their dedication to improving the quality and effectiveness of mathematics education in higher learning institutions. By leveraging technology and educational research, the authors are actively contributing to the ongoing digital transformation of university education, ensuring its relevance and resilience in the face of evolving global challenges.

¹ 元 山陽小野田市立山口東京理科大学

² 名古屋大学

^{a)} mxenn668@yahoo.co.jp

^{b)} udagawa@nagoya-u.jp

1. はじめに

筆者らは 2012 年から、高等教育機関における数学教育

向け e-Learning システムに関する共同研究を進めている。この研究活動では、第 1 筆者は数学科目の担当教師の立場として、「基礎数学」「線形代数」「微分積分学」などの担当科目において、e-Learning システムを授業内外で活用している。第 2 筆者はシステムエンジニア (SE) として、この e-Learning システムの技術的なサポートを第 1 筆者に提供している。

筆者らが運用している e-Learning システムのプラットフォームは Moodle であり、初期段階 (2012 年度頃) のシステム運用期間ではバージョン 1.9 を使用し、最近 (2022 年度) ではバージョン 4.1 を利用している。

筆者らが提供する e-Learning システムの主要な教育科目は、大学の初年次生向けの数学が中心である。そのため高度、かつ複雑な 2 次元数式 (分数・行列など複数行にわたる数式) を web サイトに表示するシステムが必要となる。これを実現するため JavaScript 数式ライブラリー「MathJax」を組み込み、HTML ソースファイルに組版システム「LaTeX」対応の TeX コマンドを使用している。加えて、Moodle の「Quiz (小テスト)」機能を使用し、数学問題の数値や関数をランダム化して出題し、記述式の解答^{*1}を求める方式を採用している。さらに、この試験システムでは Moodle サーバ群が解答された数式の正誤を自動で判断している。この学習システムの構築には、数式処理システム「Maxima」と数学オンライン評価システム「STACK」を導入している。それ故、STACK のパフォーマンスが、筆者らが提供する数学教育向けの e-Learning システムの有効性を左右する。

一方、システムではなくコンテンツ評価の観点からは、採点結果から小テストの信頼性や妥当性を分析することが必要である。容易に運用可能でありながら詳細な情報の得られる方法の 1 つとして、「S-P 表分析」が挙げられる (文部科学省 2009)。これは 2 値真偽の学習データ結果の行列 (「S-P 原表」と称する。) を特定の規則に従って並べ替え、生成される特性行列 (「S-P 表」と称する。) を用いて、学習者 (Student) と設問 (Problem) に関する 2 つの曲線 (「S 曲線」および「P 曲線」と称する。) を形成し、分析を行う方法で、「S-P table」「S-P curve theory」または「S-P chart analysis」とも呼ばれている (佐藤 1986)、(富山県 2012)、(高知県 2018)。Moodle には、S-P 表分析を行う機能は用意されていない。

本研究では以上の背景から、筆者らの運用する Moodle での e-Learning 環境の機能改善のため、システム面では STACK のパフォーマンス評価によるシステム構成の検証やチューニング、またコンテンツ面では S-P 表分析プラグインの開発や改善に取り組んできた。本稿では次節でまず S-P 表分析プラグインに関して報告し、続く 3 節で

*1 解答には表計算ソフトウェア「Excel」で利用される数式コマンドを理解すると同程度の数学的知識を必要とする。

STACK のパフォーマンス分析に関して報告する。

2. S-P 表による学習データ分析

2.1 Moodle の既存の学習データ分析

Moodle の「小テスト」機能では、さまざまな問題形式を選択できる。例えば「多肢選択問題」「○/×問題」「穴埋め問題 (Cloze)」「STACK」などが提供されている。小テストの学習データ分析には、Moodle が提供する標準的な分析方法を利用できる。例えば、「真偽値、サイズ、平均、分散などの記述統計量」と「ヒストグラム」などを含む分析結果のレポート^{*2}が Moodle の統計機能により自動的に生成される。さらに、Moodle の「評価」タブにある「統計」機能には、「小テスト情報」「問題構造分析」「問題ポジションの統計」などが含まれている。

ここで、筆者らが過去に担当した科目で行った「情報セキュリティ自己診断チェックリスト」(出典: 内閣官房情報セキュリティセンター、問題数:23 題) の問題例 (図 1) と担当科目 (受験者数:46 人) の採点結果 (図 2) を紹介する。

【1. 利用環境の設定】

新しく購入したパソコンやスマートフォンの利用環境設定時に、情報セキュリティ対策として行うべき対応のうち、最も適切な選択肢はどれでしょうか？

- 0. ウイルス対策ソフトを導入する他、起動画面にもパスワード設定を行う
- 0.2. ウイルス対策ソフトは導入せず、起動画面にパスワードを設定する
- 0.3. 信頼できる製造メーカーの製品を購入しているため、特に対策は必要ない



【2. 起動時のセキュリティ対策】

パスワードを他人から推測されにくくする工夫として最も適切な選択肢はどれでしょうか？

- 0.1. 自分と特定の友人しか知り得ない合言葉をパスワードにする
- 0.2. 個人の主観がパスワードに反映されないように、複数の友人と知恵を出し合って、複雑なパスワードを作る
- 0.3. 文字だけでなく数字や記号を織り交ぜてパスワードを作る



図 1 多肢選択問題 (2 題)

2 値採点結果の学習データ分析には、「項目応答理論」または「項目反応理論」(Item Response Theory; IRT) というテスト理論が適用されるが、この理論を用いるためには外部のソフトウェアが必要であり、操作も安易ではない。

そこで次の小節では、筆者らによって開発された使い勝手が良い「S-P 表分析」プラグインについて紹介する。

2.2 開発した「S-P 表分析」プラグイン

この小節では、2019 年に MAJ の「開発と研究の助成金支援」に「S-P 表プラグインの開発と実践」の研究テーマで応募し、採択された後に開発された「S-P 表分析」プラグイン「moodle-gradereport_sptable」を紹介する。このプ

*2 この分析法は「古典的テスト理論」として知られている。



図 2 採点結果表 (一部)

ログインは、「Moodle Moot Japan 2020」で「S-P 曲線」を生成する初期バージョンについて研究成果として紹介している(亀田・宇田川 2020)。その後継続的に改善され、「注意係数(学習者)×得点」や「注意係数(設問)×正答率」を示す散布図などの分析を自動で生成できるようになった。これらは「Moodle Moot Japan 2024」で紹介された最新機能である(亀田・宇田川 2024)。

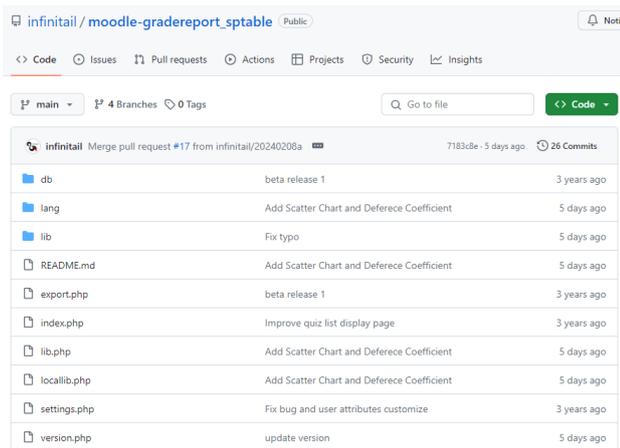


図 3 GitHub リポジトリ (イメージ)

このプラグインは、以下の GitHub リポジトリからダウンロードできる:「https://github.com/infinitail/moodle-gradereport_sptable」(図 3)。

2.3 開発した「S-P 表分析」プラグインの使用法

この小節では、開発されたプラグインを Moodle (Ver.4.1) サイトにおける<教員>ロールをもつユーザがどのように使用するかについて解説する。

以下の手順に従って S-P 表分析を実行する。

- (1) 「評価」タブの「評価者レポート」メニューから「S-P 表分析」機能を選択する (図 4)
- (2) 「S-P 表分析」リストから分析対象の小テストを選択する (例:「情報セキュリティ...」)
- (3) 学習データ分析結果を含む Excel ファイルを端末に保

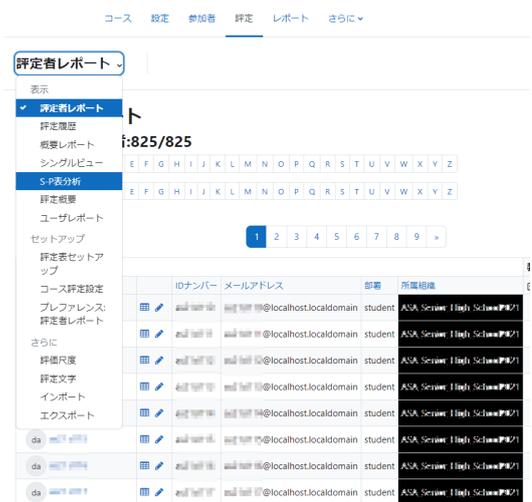


図 4 「評価レポート」ページの操作イメージ

存する (ここでは「sptable...」である)

- (4) Excel ファイルを開くと、4つのシートで構成されている
- (5) 第1シート「S-P 表分析」では「S-P 曲線」を含む分析結果が表示される (図 5)、なお詳細な学習データ値 (受験者に対する「得点」「正答率」「C.S. (注意係数)」、設問に対する「正解数」「正答数」「C.P. (注意係数)」)を表 A.1 に掲載する。ここで正解時の場合に「1」が、誤答時の場合に「0」がそれぞれ割り振られている
- (6) 第2シート「ScatterStudent」では「散布図:注意係数(学習者)×得点」の分析結果が表示される (図 6)
- (7) 第3シート「ScatterProblem」では「散布図:注意係数(設問)×正答率」の分析結果が表示される (図 7)
- (8) 第4シート「Analytics」には、「受験者数 (Number Student)」「設問数 (Number Question)」「平均点 (Answer Rate)」「関連するサイズ指数 (M index)」「関連する比率 (DBM)」「差異係数 (Deference Coefficient)」といった複数の分析指標が出力されている

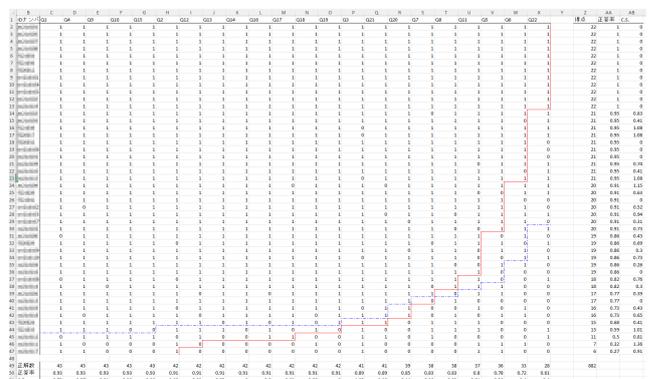


図 5 「S-P 表分析」シートのスクリーンショット

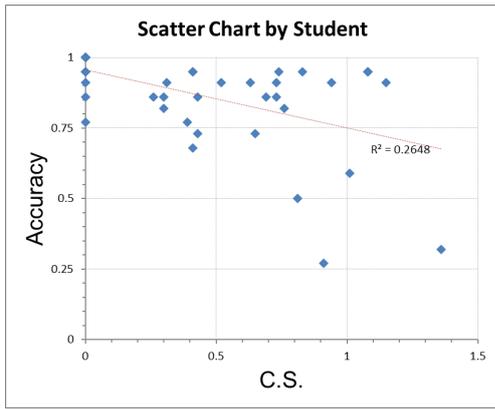


図 6 散布図：注意係数（学習者）× 得点

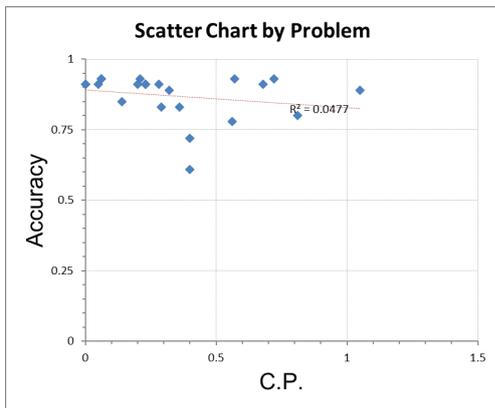


図 7 散布図：注意係数（設問）× 正答率

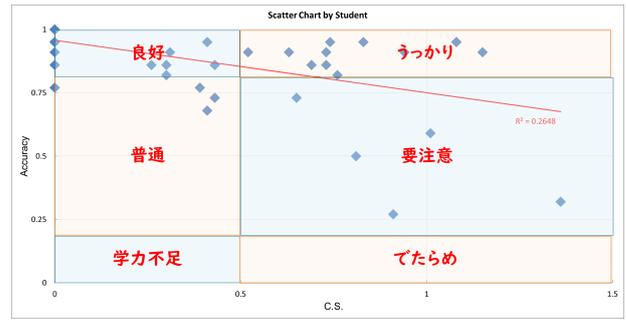


図 8 アレンジ散布図：注意係数（学習者）× 得点

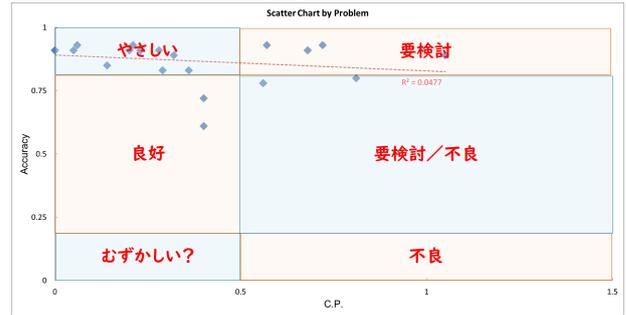


図 9 アレンジ散布図：注意係数（設問）× 正答率

2.4 分析結果「散布図」のアレンジ例

この節の最後に、「S-P 表分析」プラグインから得られた 2 つの散布図を表計算ソフトウェア「Excel」で視覚的に分かりやすくアレンジした例を紹介する（図 8, 9）。

なお、分析の詳細については、参考文献（富山県 2012）に記載されているので、そちらを参照してほしい。

- (1) 散布図はシンプルなグラフ形式であり、Excel での編集が可能である。基本的な図形挿入機能で編集したアレンジ例を図 8（学習者向け）と図 9（設問向け）で示す
- (2) 散布図 8 の分析結果によると、【学力不足】や【でたらめ】とされる学習者はいなく、【要注意】とされる学習者が、全学習者 48 人の中に 5 人いる
- (3) 散布図 9 の分析結果によると、【むずかしい?】や【不良】の両カテゴリに該当する設問はなく、【要検討/不良】とされる設問が、全設問 23 題の中に 2 題含まれる
- (4) 「差異係数 (Difference Coefficient)」*3 の値は約 0.45 で、これは特に注意の必要な問題形式ではないと判断できる

以上である。

*3 差異係数について、問題形式（例：「試験」「アンケート」など）に従って注意すべき範囲が設定されている。

3. STACK のパフォーマンスについて

3.1 最初に

STACK は PHP と Maxima を使用して動作する Moodle 用プラグインである。筆者らはこのプラグインを利用して、大学における数学授業の効果的な運営を目指している。これまでに、他大学の STACK 管理者との情報交換を通じて、運用中に得られた知見やさまざまな意見を確認することができた。その際に STACK の処理速度についての認識に差異があることに気がついた。

筆者らは、参考文献（亀田・宇田川 2013）では、システム構成「Moodle 1.9 + STACK 2.0」の e-Learning システムにおける「負荷分散、および Optimised の高速化」について、また参考文献（亀田・宇田川 2015）では、システム構成「Moodle 2.8 + STACK 3.3」の e-Learning システムにおける「CLISP と SBCL で LISP コンパイラによる性能比較と Maxima Pool」について、それぞれ研究結果を論述している。引き続き本稿では、システム構成「Moodle 4.3 + STACK 4.6」の e-Learning システムにおける研究結果を報告していく。表 1 に文献別調査システムのバージョンについてまとめた。

表 1 文献別調査システムのバージョン

文献	Moodle Ver.	STACK Ver.
亀田・宇田川 2013	1.9	1.0
亀田・宇田川 2015	2.8	3.3
本稿	4.3	4.6

3.2 STACK のパフォーマンスに関する問題の調査

STACK は、Moodle の小テスト機能に新たな問題タイプを追加し、実践的な数学問題の出題を可能にしている。具体的には STACK の Moodle 用モジュールから Maxima を呼び出し、その評価結果を解答にフィードバックする。Maxima の処理には比較的時間が掛かるため、その処理時間を短縮する方法と効果が、参考文献 (亀田・宇田川 2013) (pp.22-27), (亀田・宇田川 2015) (以下、この2つの参考文献を「先行研究」という。) で説明されている。

今回、実際の数学小テストの実施時に STACK が要する処理時間とその中で Maxima が占める割合を調査した。

プログラム実行時の処理時間を調査するために「プロファイラ」と呼ばれるソフトウェアが使用される。ただし、プロファイラは詳細な実行情報を提供する一方で、処理時間に影響を与えることがある。今回は、Tideways (<https://tideways.com/>) というプロファイラを使用した。Tideways は PHP モジュールとして動作し、実行結果を Tideways の提供する「SaaS」*4 プラットフォーム上で確認できる。

使用したサーバのシステム環境は表 2 に示す。また、プラットフォームとして「IaaS」*5 である「Oracle Cloud Infrastructure (OCI)」を利用した。OCI では、「Simultaneous Multi-Threading (SMT)」対応 CPU を使用する場合、物理コア単位で 2 つのスレッドが割り当てられる (要確認)、「1 OCPU」が一般的な IaaS の「2 vCPU」に相当する。① Moodle および データベース (DB) が動作しているインスタンスと、② MaximaPool 用のインスタンスを含む 2 つのインスタンスを準備し、それぞれに環境を構築した。両インスタンスは同一リージョン内に設置されており、インスタンス間の平均 ping 値は 0.213 ミリ秒であった。

第 1 に、Tideways の導入がパフォーマンス、特に実行時間にどの程度影響するかを調査した。具体的には、後述する「線形代数」科目の小テストにおける開始時の処理にかかる時間を 5 回測定し、キャッシュの影響を排除するため最初の 2 回分を破棄して、残り 3 回のうちの中央値を Tideways の導入前後で比較した。この比較検証は httpd のログに記録された処理時間を利用し、導入前で 3,313.091 ミリ秒が、導入後で 3,212.268 ミリ秒がそれぞれ計測できたことで、Tideways の導入がパフォーマンスに与える影響は無視して良いと判断した。Tideways 導入後の方がパフォーマンスが良いのは、実行時間のゆらぎによるもので

*4 Software as a Service の略称であり、サービス提供事業者 (サーバ) 側で稼働しているソフトウェアを、インターネットなどのネットワークを経由して、ユーザーが利用できるサービスである。

*5 Infrastructure as a Service の略称であり、サーバ、ストレージ、CPU、メモリ、ネットワークなど、コンピューターシステムの構築に必要なインフラをインターネット上で提供するサービスである。

表 2 検証サーバ (VM.Standard.E4.Flex) のシステム構成

インスタンス	機能・要素
CPU	4 OCPU (8 vCPU)
Memory	4 GB
OS	Oracle Linux Server release 8.9
Moodle	4.3.3
STACK	4.6.6
PHP	8.0.30
MySQL	8.0.36
Maxima	5.43.0
LISP	SBCL 1.2.16
OpenJDK	11.0.22.0.7
Tomcat	9.0.65

あると考える。

今回検証した小テストは、3 つのカテゴリ「練習」「微分方程式」「行列」に属する問題を使用した。それぞれの問題 (一部分のスクリーンショット) は図 10、図 11、図 12 を参照してほしい。



図 10 カテゴリ「練習」の問題 (一部)

検証した小テストの回答欄数を、カテゴリ別および大問別でまとめた結果を表 3 に示す。なお行列の問題に回答する場合、成分の数ではなく 1 つの回答欄としてカウントする。

表 3 検証した小テストのカテゴリ別・大問別の回答欄数

カテゴリ	大問数	大問毎の回答欄数										
練習	10	3	3	3	3	3	3	3	3	5	5	
微分方程式	2	6	2									
行列	11	1	6	7	3	9	2	4	6	3	5	7

次に、今回の調査対象は、処理に比較的時間が掛かる「小テストの開始時 (startattempt.php) における所要時間」

【微分方程式】 変数分離形 ❗ 問題にはテストが変形が不足しています。

【問題】 微分方程式に対する問いに答えよ。

1. 1階微分方程式の一般解 $f(x, C)$ を求めよ。ただし任意定数は C とする。

$$y' + 2xy = 0 \Rightarrow \text{一般解: } y = f(x, C) \Rightarrow f(x, C) = A_1.$$

$A_1 =$

2. 初期値問題の解すなわち特殊解 $f(x)$ を求めよ。

$$\frac{dy}{dx} - y - 1 = 0 \quad (x = 0, y = 2) \Rightarrow \text{解: } y = f(x) \Rightarrow f(x) = A_2.$$

$A_2 =$

3. 境界値問題の解すなわち特殊解 $f(x)$ を求めよ。

$$\frac{d^2y}{dx^2} = 6x \quad (x = 1 \text{ のとき } y = 1, \quad x = -1 \text{ のとき } y = 3) \Rightarrow \text{解: } y = f(x) \Rightarrow f(x) = A_3.$$

$A_3 =$

4. 同次形の微分方程式について、一般解 $f(x, y, C)$ 及び特異解 $g(x)$ を求めよ。ただし 0 でない任意定数を C とする。

$$\frac{dy}{dx} = \frac{2xy}{x^2 - y^2} \Rightarrow \begin{cases} \text{一般解: } 0 = f(x, y, C), \\ \text{特異解: } y = g(x). \end{cases} \Rightarrow \begin{cases} f(x, y, C) = A_4, \\ g(x) = A_5. \end{cases}$$

$A_4 =$

$A_5 =$

図 11 カテゴリ「微分方程式」の問題（一部）

掃き出し法 ❗ 問題にはテストが変形が不足しています。

【問題】 次の連立1次方程式について答えよ。

$$\begin{cases} 2x + 3y - z = 1 \\ 2x + y + 4z = 6 \\ -3x + 6y - z = 6 \end{cases}$$

1. 連立1次方程式に対応した係数行列 A 、拡大係数行列 $A' = [A | \mathbf{b}]$ を求めよ。|

$$A = A_1, \quad A' = A_2$$

$A_1 =$

$A_2 =$

2. 未知数の個数 n 、2つの行列の階数 rank を求めよ。

$$n = A_3, \quad \text{rank}(A) = A_4, \quad \text{rank}(A') = A_5.$$

$A_3 =$

$A_4 =$

$A_5 =$

3. 次のように消去法（すなわち掃き出し法）を用いて、連立方程式の解 \mathbf{x} を求めよ。ただし E は3次単位行列である。

$$A' \xrightarrow{\text{行基本変形}} [E | \mathbf{x}].$$

図 12 カテゴリ「行列」の問題（一部）

(以下、「開始時」と記す。)、および「小テストの回答送信時 (processattempt.php) における処理時間」(以下、「回答時」と記す。)) を選択した。

ただし、これらの小テストは、<学生>ロールでの受験ではなく、<教師>ロールでのプレビューとして実行している。また STACK の設定では Maxima は「Linux (Optimised)」(以下、「Optimised」と記す。)) を使用し、数式処理 (Computer Algebra System; CAS) キャッシュは無効に設定した。

Tideways の評価と同様、各小テストを 5 回プレビューし、最後の 3 回の中央値を使用した。プレビュー中の全回答欄には一律「1」を入力している。

プロファイラの結果を 3 つの表で示す。表 4 が実行時間、表 5 が呼び出し回数、表 6 が占有率である。ただし、Maxima の実行時間の占有率については、Tideways が全

ての処理についてではなくサンプリングして分析結果を返すため、後半 3 回の実行でサンプリング対象となった結果の一つを使用している。

表 4 カテゴリ別実行時間 (STACK 中の Optimised)

カテゴリ	小テストの実行時間 (ミリ秒)	
	開始時	回答時
練習	3,229	5,323
微分方程式	845	1,318
行列	6,356	10,653

表 5 カテゴリ別呼び出し回数 (STACK 中の Optimised)

カテゴリ	Maxima の呼び出し回数	
	開始時	回答時
練習	54	96
微分方程式	12	12
行列	104	168

表 6 カテゴリ別占有率 (STACK 中の Optimised)

カテゴリ	Maxima の占有率 (%)	
	開始時	回答時
練習	94	97
微分方程式	89	91
行列	92	95

参考として、Tideways のスクリーンショットを示す。図 13 が「STACK 中の Optimised のサマリ」、図 14 が「STACK 中の Optimised のタイムライン」に関するそれぞれの画像である。これらの結果から本論文で利用しているデータの一部を取得している。サマリでは PHP スクリプト中での各ルーチンの実行回数や DB アクセス、外部プログラムの呼び出しの回数や累計時間などが表示されており、タイムラインではそれらの実行時間が時系列順にプロットされている。一般的にはこの分析結果から処理全体におけるボトルネックの特定を行うことになる。

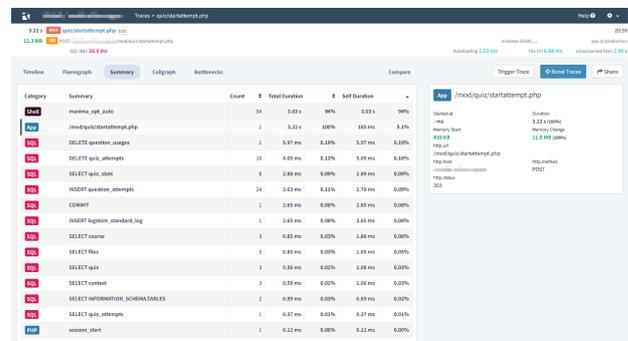


図 13 STACK 中の Optimised のサマリ

これらの事実から、実行時間の大部分が Maxima (maxima_opt_auto) の処理に費やされており、これが全

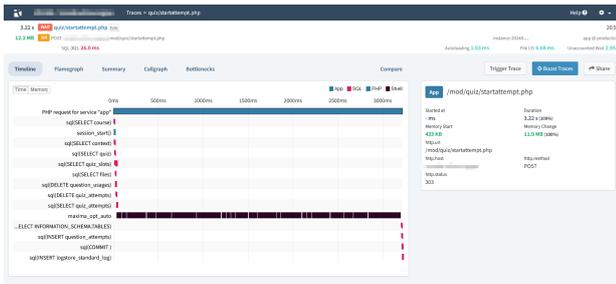


図 14 STACK 中の Optimised のタイムライン

体のパフォーマンスに大きな影響を与えていることがわかる。Maxima の実行それぞれはごく短い時間だが、回数が多いため全体として長い処理時間となっている。逆に STACK による CAS テキストの生成や DB へのアクセスについて、どれほど効率化を図ったとしても全体的なパフォーマンス向上にはほとんど影響がないと言える。

Maxima の呼び出し回数は、小テストの構成により、問題数や入力欄の数から単純に計算することができなくなっている。また、本小テストでは利用していないポテンシャルレスポンスツリーも Maxima の呼び出し回数に影響を及ぼすことがある。

3.3 MaximaPool の使用環境

STACK の処理において、Maxima の実行時間を減らす方法として、MaximaPool (https://github.com/math5/stack_util_maximapool) が STACK の開発者により用意されている。MaximaPool は Java で書かれたプログラムで、ウェブ API を通じて Maxima にアクセスできるサービスを提供し、Maxima の処理時間を短縮することを目的としている。先行研究では、STACK 中の Optimised と MaximaPool の間で大きな処理時間の差は見られず、STACK 中の Optimised が起動時にプロセス生成を必要とする一方で、MaximaPool はネットワークを介する遅延があるため、顕著な差はないであろうとされている。

この小テストも、3つのカテゴリごとに小テストをレビューで受験した。掲載は省略するが、Maxima の実行回数は前掲の Optimised と同じであった。

MaximaPool 使用時の Tideways のスクリーンショットを以下に示す。図 15 が「MaximaPool 使用時のサマリ」、図 16 が「MaximaPool 使用時のタイムライン」である。

MaximaPool 使用時のプロファイラから得られる結果を表 7 にまとめる。

表 7 カテゴリ別実行時間 (MaximaPool 使用時)

カテゴリ	小テストの実行時間 (ミリ秒)	
	開始時	回答時
練習	1,676	2,493
微分方程式	477	644
行列	3,602	6,275

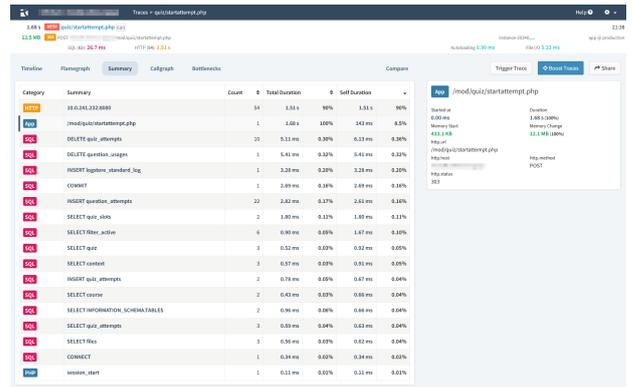


図 15 MaximaPool のサマリ

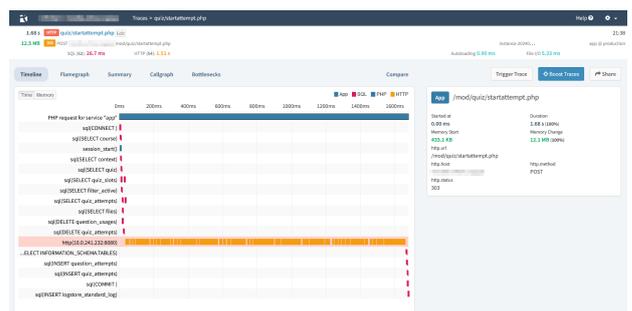


図 16 MaximaPool のタイムライン

表 4 と表 7 を比較して、MaximaPool 使用時の STACK の実行時間が、STACK 中の Optimised の約半分に短縮されたことが明確に示されている。

先行研究では大きな差が無いとされていたが、実際は大幅な高速化を図ることができると予想される結果となった。この点について先行研究と比較しようと考えたが、既に 10 年近く経っており同等の環境を用意する事が不可能であるため、当時の環境と同じ条件での比較は断念した。

3.4 CAS キャッシュ

STACK には Maxima の処理を高速化するため、データベース (DB) を利用したキャッシュ機能が組み込まれている。これは、Moodle のデータが保存されている DB を利用して実現されている。なお、ドキュメントに記述はないが、Moodle が利用しているものとは異なる DB を利用することも可能になっている。

具体的には、CAS キャッシュは以下の手順で動作する。

- (1) ユーザが Moodle 上で、STACK の問題に何らかのアクションを起こすと、プロセスが開始される
- (2) STACK モジュールは、Maxima で処理するための CAS テキストを生成する
- (3) CAS キャッシュが有効な場合、DB 内で CAS テキストに対応するレコードを検索する
- (4) 次に、Maxima は CAS テキストを実行し、その結果を出力する
- (5) CAS キャッシュが有効な場合、CAS テキストと Max-

ima の実行結果と一緒に DB に保存する

(6) 最後に、STACK モジュールが CAS テキストの実行結果を処理し、その結果を Moodle 上で表示する

固定された結果を求める問題を解く際には、初めての場合は Step 1 から 6 まで全て実行する必要があるが、2 回目以降では Step 4 と 5 を省略することができる。CAS キャッシュが無効の場合は、Step 3 と 5 も省略される。Maxima の処理には時間がかかるが、DB からレコードを取得する時間は非常に短いため、全体の処理時間の短縮が可能である。これは、CAS テキストをハッシュ化した値をインデックスとして使用し、検索を効率化することで実現している。

3.5 CAS キャッシュの問題点

CAS キャッシュは Moodle の DB 内の「qtypestackcachecache」テーブルに随時記録される。「Least Recently Used (LRU)」方式の管理が行われていないため、運用の仕方によってはテーブルのレコード数が無限に増大する可能性がある。また、CAS テキスト自体のサイズが大きいため、それぞれのレコードも大きくなり、33 万行程度でもテーブルのダンプが 1GB を超えるようなサイズとなる。この 33 万行は筆者らが独自に運用している Moodle で半年にも満たない期間に記録されていた。

このテーブルの肥大化の一因は、筆者らが STACK の「デプロイ (Deploy)」機能を使用せず、完全にランダムな問題を作成したため (図 17) である。例えば、図 17 のような変数を持つ問題であれば、 $40 (= 5 \times 4 \times 2)$ パターンの CAS キャッシュでカバーできると思われるが、Moodle の仕様により、実際には 100 万パターンの疑似乱数のシードが利用される。そのため CAS キャッシュが有効に機能せず、CAS テキストが生成されるたびに DB に記録されてしまっていた。

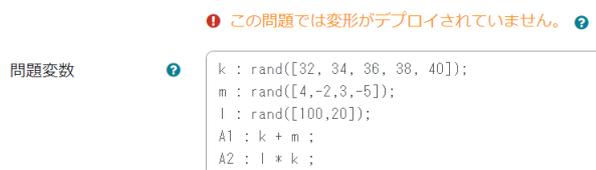


図 17 カテゴリ「練習」の小テスト第 1 問題の編集ページにある「問題変数」ブロック

STACK のデプロイ機能では、事前に複数の疑似乱数のシードにより問題を解析し、DB にキャッシュし、小テストで同じ疑似乱数を利用することで、CAS キャッシュの利点を最大化することが可能となっている。このデプロイを有効に活用することで、レコード数を抑え、CAS キャッシュの恩恵にあずかることもできるようになると思われる。ただし、デプロイには欠点も存在し、問題作成時に必要な疑似乱数を生成する必要があり、作成者は全パターンの処理が完了するまでオンラインで待機する必要がある。

3.6 課題

この検証結果から、様々なケースにおいて MaximaPool の利用が効果的であることが確認できた。ただし、今回の検証では STACK 中の Optimised は Moodle と同一のサーバ上で動作しており、MaximaPool は異なるサーバで動作している点が相違している。そのため、処理時のサーバの負荷として Maxima Pool 側が有利となっている。また、低負荷時には MaximaPool が速度向上に寄与するが、多数の利用者が同時に小テストを受験するような高負荷において、STACK 中の Optimised と MaximaPool のどちらがより効果的かについてはまだ比較が行われていない点が課題である。

参考文献

- 亀田真澄、宇田川暢 (2013). Moodle, TeX, STACK による数学の e ラーニングの取り組み、日本 Moodle 協会全国大会発表論文集 1 巻 (2013), pp.22-27, https://moodlejapan.org/file.php/1/2013_Moot_files/MoodleMoot2013Proceedings.pdf (確認日: 2024 年 4 月 26 日)
- 亀田真澄、宇田川暢 (2015). Moodle による数学基礎 e-Learning の取り組み、日本 Moodle 協会全国大会発表論文集 (2015), pp.11-16, https://moodlejapan.org/pluginfile.php/4690/mod_resource/content/6/Proceedings2015.pdf (確認日: 2024 年 6 月 22 日)
- 亀田真澄、宇田川暢 (2020). Moodle を利活用した数学オンライン記述式試験の実践報告、Moodle Moot Japan 2020, <https://moodlejapan.org/course/view.php?id=83> (確認日: 2024 年 4 月 25 日)
- 亀田真澄、宇田川暢 (2024). Moodle におけるデジタルライゼーション化へのアプローチ ～小テストに対する学習分析ツールと作問のコンテンツ作り～、Moodle Moot Japan 2024, <https://moodlejapan.org/course/view.php?id=99> (確認日: 2024 年 4 月 25 日)
- 高知県 (2018). S-P 表作成ワークシート, <https://www.pref.kochi.lg.jp/soshiki/310304/download-sp.html> (確認日: 2024 年 2 月 13 日)
- 佐藤隆博 (1986). S-P 表の入門、教育実践文庫 ③、明治図書
- 富山県中学校教育研究会 (2012). S P 表分析について < 1st Step～はじめの一步～>, <https://www.ktkk.ico.bz/upfile/spp.pdf> (確認日: 2024 年 4 月 26 日)
- 文部科学省 (2009). 学校/学級別解答状況整理表 (S-P 表) の活用方法について, https://www.mext.go.jp/a_menu/shotou/gakuryoku-chousa/1409618.htm (確認日: 2024 年 2 月 8 日)

付 録

A.1 S-P 表分析の実践例

表 A.1 S-P 表分析例

ID	Q1	Q4	Q9	Q10	Q15	Q2	Q12	Q13	Q14	Q16	Q17	Q18	Q19	Q3	Q21	Q20	Q7	Q8	Q11	Q5	Q6	Q22	得点	正答率	C.S.	
ID01	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID02	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID03	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID04	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID05	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID06	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID07	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID08	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID09	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	22	1	0	
ID13	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	21	0.95	1.08	
ID14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	21	0.95	0.83	
ID15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	21	0.95	0.41	
ID16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	21	0.95	0	
ID17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	21	0.95	0	
ID18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	21	0.95	0.41	
ID19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	21	0.95	0.74	
ID20	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	21	0.95	1.08	
ID21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	21	0.95	0	
ID22	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	21	0.95	1.08	
ID23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	20	0.91	0.63	
ID24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	20	0.91	0	
ID25	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	20	0.91	0.52	
ID26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	20	0.91	0.31	
ID27	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	20	0.91	1.15	
ID28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	20	0.91	0.94	
ID29	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	20	0.91	0.73	
ID30	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	19	0.86	0.69	
ID31	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	19	0.86	0.73	
ID32	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	19	0.86	0.43	
ID33	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	19	0.86	0.3	
ID34	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	19	0.86	0.26	
ID35	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	19	0.86	0	
ID36	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	18	0.82	0.76
ID37	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	18	0.82	0.3	
ID38	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	0	0	17	0.77	0.39	
ID39	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	17	0.77	0	
ID40	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	0	0	1	0	1	0	16	0.73	0.43	
ID41	1	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0	1	0	1	1	0	16	0.73	0.65	
ID42	1	1	1	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	0	0	15	0.68	0.41	
ID43	1	1	1	0	0	1	1	1	0	1	0	1	0	1	0	0	1	1	1	0	0	1	13	0.59	1.01	
ID44	0	1	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1	0	0	0	0	1	11	0.5	0.81	
ID45	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	7	0.32	1.36	
ID46	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	6	0.27	0.91	
正解数	43	43	43	43	43	42	42	42	42	42	42	42	42	41	41	39	38	38	37	36	33	28	882			
正答率	0.93	0.93	0.93	0.93	0.93	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.89	0.89	0.85	0.83	0.83	0.8	0.78	0.72	0.61				
C.P.	0.72	0.57	0.21	0.06	0.06	0.68	0.23	0.05	0	0.2	0.23	0.28	0	1.05	0.32	0.14	0.36	0.29	0.81	0.56	0.4	0.4				

大学初修生物学におけるオンラインテスト実施方法の変更による 成績評価改善の検証

中馬いづみ・得字圭彦・斉藤準

帯広畜産大学人間科学研究部門

帯広畜産大学基盤教育の必修科目である生物学概論を、Moodle を用いて運営している。受講生は1年生約250人と過年度履修生を合計して260人前後である。本科目の成績は、講義前の1週間で実施される予習課題、講義後の1週間で実施される復習課題に、中間試験および期末試験を加えて合計100点満点として評価する。合格点とする60点以上を得点した学生について、2020年度以降、平均点が高くばらつきが小さい傾向が続いた。そこで2023年度は、これまで何回も受験可能としていた予習・復習課題を1回のみ受験可能となるよう設定変更し、中間試験を廃止し期末試験のみ実施した。この成否について、平均点と点数の分布、および小テストの受験結果やMoodleの標準ログを利用して一括取得した各種テストの解答時間等の経年変化から検討した。その結果、点数の分散が増加し平均点が減少したことにより、識別率が増加した。評価の信頼性は十分であり、妥当性も複数の指標から概ね担保された。解答時間と成績の間に直接的な相関は認められなかった。課題として、本実践の持続可能性や学習効果の向上のためには、問題バンクの拡充が必要であることが示唆された。

Verification of the improvement in grading methods through modified online test administration for an undergraduate introductory Biology course

IZUMI CHUMA, YOSHIHIKO TOKUJI, JUN SAITO

Obihiro University of Agriculture and Veterinary Medicine

The course "Introduction to Biology," a required course for basic education at Obihiro University of Agriculture and Veterinary Medicine, is facilitated using Moodle. The total number of students in the course is about 260, including approximately 250 first-year students and students who took the course in previous years. Grades for this course are determined by the total score out of 100 points, which is the sum of a preparatory test given one week before each lecture, a review test given one week after the lecture, a mid-term exam, and a final exam. Among students who scored 60 or more points, which is considered a passing score, the trend of high average scores and low variance continued after the school year of 2020. Therefore, in the current (2023) school year, the settings were adjusted to allow students to take each of the preparation/review tests only once, while the mid-term exam was abolished, leaving only the final exam. Consequently, the variance of the scores increased and the mean score decreased, resulting in improved discrimination of the students' performance. The reliability of this assessment was sufficient, and its validity was generally assured. No direct correlation was found between the time taken to answer the questions and the scores. While these findings imply the enhanced effectiveness of the grading methods, the results also suggest that the question bank needs to be expanded for the sustainability of this practice and the quality of student learning.

1. はじめに

筆者らはこれまでに、帯広畜産大学における生物学教育のMoodleを用いた評価方法を検討してきた。中馬ら(2022)および中馬ら(2023)では、約120名の学生が毎年受講する生物学実験の授業において、特にテクニカルライティングの指導とその評価に対し、TAとの協働によるMoodleのルーブリックを用いた実験ノートの採点が有効であることを明らかにした。今回、これより倍以上となるおよそ260名の学生が受講し、実習形式ではなく講義形式の基礎科目である「生物学概論」について、Moodleによる成績評価の方法を検討した。

本学は日本の国立大学で唯一の獣医農畜産系単科大学で、

畜産学部のみを有するが、教育研究領域としては生物学の幅広い分野を取り扱っている。この領域多様性を反映して入試制度およびそれによる入学者も多様であり、本学の学生の中には、生物学が好きでこれを得意とする学生がいる一方で、高校時代に大学受験のための生物学の学習を行わなかったことを理由に生物学に苦手意識を持つ学生もいる。我々が教養生物学の教育を行う上で必要とされることは、こうした多様な学生個々が生物学に興味を持つきっかけをつくり、専門教育の共通基盤となる幅広い知識を身につけさせることである。これらのことから、生物学概論の評価のための試験には、特定分野に特化し一部の学生のみが対応可能であるような高度な記述を要する問題等ではなく、基礎的な多肢選択問題をさまざまな分野から偏りなく多数

出題している。これにより、生物学を不得意とする学生も学習への取り組み次第で得点が可能となり、得意とする学生も学習成果に応じて高得点を目指すことができ、かつ分野の幅広さにも対応できるという点で、本科目の成績評価として適するものと考えられる。実際、多肢選択問題は学習の幅広さや多様な学習水準のテストに用いることができる (Davis 1993)。

Moodle の小テストの結果の分析に関する先行研究としては、多肢選択問題の小テストに関し項目反応理論を含むテスト理論の観点から分析を行った和田 (2012)、野崎・村川 (2019)、学生をつまづきの検出を行った加藤 (2012) などがある。また、大人数の大学基礎科目の成績評価を Moodle で行った先行研究において、試験等の成績だけでなく、Moodle から得られる予習・復習課題の評点やその学習時間・学習時刻等の指標に着目することにより、定量的・客観的な実践評価が可能であることが示されている (齊藤 2020, 齊藤 2021b)。また、Moodle を用いたオンライン試験において、制限時間に対する分量の割合を上げ、受験者ごとにランダム化した計算問題を用いることで、正規分布の意味で適切な試験が実行可能であることが報告されている (齊藤 2021a)。

本研究では、これらの先行研究の対象とは異なる生物学科目において、制限時間に対して多数の多肢選択問題からなる試験や、予習・復習課題の評点による成績評価の方法について、試験の解答時間等の指標も考慮しつつ検証する。

2. 科目概要

2.1 科目内容

帯広畜産大学において、生物学概論は基盤教育の必修科目に位置づけられ、1 年次後期に開講している。1 学年の人数は共同獣医学課程 40 名、畜産科学課程 210 名で、これに過年度履修生 10 名程度を加えた 260 名前後の学生が例年受講する。本学のほとんどの学生が理系であり、また獣医農畜産系という専門性から、何らかの生物または生物由来の物質や現象を取り扱う学生が多いこともあり、生物学概論では、すべての学生が今後生物学の延長として専門分野を学ぶことを想定した内容で講義を行っている。

本科目は 2018 年度より中馬が担当しており、初年度は『カラー図解 アメリカ版 大学生物学の教科書 (講談社ブルーバックス) 1~4 巻』を指定した。2019 年度から現在までの指定教科書は『キャンベル生物学 原書 11 版』(丸善出版) で、講義で利用するのはこの一部であるが、講義終了後にも事典として、各学生が専門分野の学習の際の導入として利用できるよう、本書を選定した。

講義内容は表 1 のとおりである。科目担当者は 6 名おり、このうち中馬と得字が教科書に沿った基礎的な学習内容を担当し、その他の教員は応用としてそれぞれの専門分野への導入的な学習内容を扱う。試験範囲は表 1 でアスタリスクを付した中馬・得字の講義内容としている。なお、表 1 の講義内容は 2023 年度のものであるが、開始当初より概ね同様の内容で構成している。試験は 2022 年度までは中間試験と期末試験を行っていたが、2023 年度は後述の理由により期末試験のみ実施した。

表 1 2023 年度生物学概論の内容

	日付	担当者 (内容)	教科書の章
1	10/6	中馬 (オリエンテーション)	
2	10/13	得字 (動物の栄養) *	41
3	10/20	近藤 (人体の宇宙)	40 (参考)
4	10/27	中馬 (細胞と細胞周期) *	6, 12
5	11/10	中馬 (減数分裂と有性生活環) *	13, (14)
6	11/17	中馬 (メンデル遺伝学とその分子機構) *	14, 15, 16
7	11/24	中馬 (遺伝子発現の機構) *	17, 18
8	12/1	得字 (細胞呼吸と発酵) *	9
9	12/8	得字 (光合成) *	10
10	12/15	山内 (昆虫の博物学)	33 (参考)
11	12/22	中馬 (系統と生命の樹) *	26
	1/5	休講	
	1/12	大学入学共通テスト準備のため休講	
12	1/19	中馬 (動物・植物の病原体との戦い) *	43, 39.5
13	1/26	中馬 (生態学の入門と生物圏) *	52
14	2/2	宮崎 (森林生態学)	
15	2/9	紺野 (生態学)	
16	2/16	期末試験	

2.2 Moodle の使用と成績評価方法

講義に関する資料の配布、各回の講義前後の小テスト、試験はすべて Moodle を使用した。講義の各回の内容はひとつのトピックにまとめ、基本的に、講義資料 (講義スライドを PDF ファイルに変換したもの)、予習課題 (1 週間前の講義終了後から講義開始前までに提出する)、復習課題 (講義終了後から次回講義開始前までに提出する)、これと必要に応じて動画や PDF ファイル等の参考資料を提供した。これらのうち、予習課題と復習課題を課すのは表 1 のアスタリスクを付した全 10 回である。予習課題および復習課題では、基本的に 1 回あたりの制限時間を 30 分として多肢選択問題を 5 題出題し、10 満点とした。これらは、成績にも使用される評価対象として課すことにより各授業段階における学習を促すとともに、学生に対する形成的評価 (有馬 2021) としてフィードバックとリフレクションの機会を与え学習を深めることをねらいとするため、一定の頻度が必要と考えた。このようにして展開した講義に加え、2022 年度までは中間試験および期末試験を行い、2023 年度は期末試験のみを実施した。すべての課題と試験は問題および選択肢をシャッフルした。

成績評価は、予習課題と復習課題の合計と、中間試験と期末試験の合計がおおよそ 1:1 となるよう配分し、100 点満点とした合計点を評点としている。ただし、点数の分布と平均点を考慮し、60 点未満となった学生の一部も合格点の 60 点となるよう調整した場合もある。

3. 2018 から 2022 年度の成績の推移と課題

2018~2022 年度の成績の推移を図 1 に示した。このグラフでは、合格者のみの得点 (60~100 点) を表示した。予習課題と復習課題は受験可能回数を無制限とし、中間試験と期末試験は受験可能回数を 1 回に設定したこともあり、平均点 (折れ線) は 2018 年度と 2019 年度では 80 点以上、

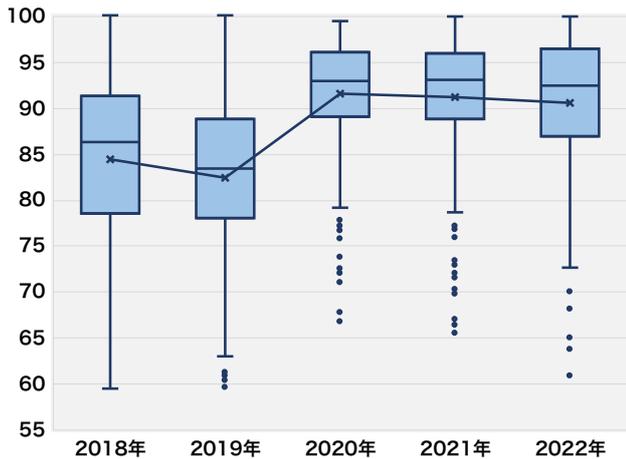


図1 2018～2022年度の成績の推移

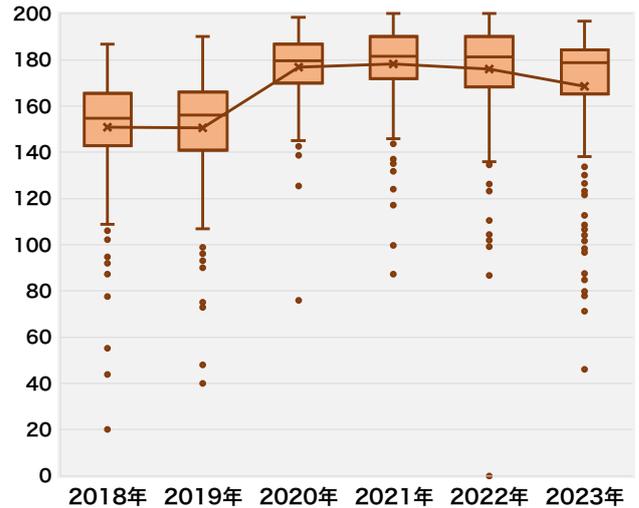


図3 中間試験と期末試験の合計点数の比較

2020年以降は90点以上となった。教科書の変更は2019年度からで、それに伴い平均点はやや下がったが、その後は平均点が大きく上昇し、点数のばらつきが減少した。

成績評価としてはある程度の点数差すなわち分散を生じさせ、また天井効果(豊田 2000)を避けるために平均値があまり高くないように下げ、学生間の成績の識別性を担保することが求められる。そのためには、Moodleでの出題条件を変更する必要があると考えた。

4. 2023年度の成績評価

4.1 2023年度の変更点

過去3年間の平均点が高く、点数のばらつきが小さい傾向を解消することを目的に、2023年度は以下の点について変更した。予習課題および復習課題は制限時間30分のまま、受験可能回数を無制限から1回に変更した。試験は中間試験を廃止し期末試験のみ実施した。これまで中間試験と期末試験はともに約60問を制限時間30分で解答する形式であったが、これらの試験を1回にまとめ制限時間を短縮し、128問を50分で解答することとした。問題の内容は一部修正したが基本的には前年度と同じ問題を出題した。

4.2 変更による点数の変化

予習課題と復習課題の受験可能回数を1回にした影響を検討し、その結果を図2に示した。2022年度の課題のうち、第5回の予習課題(22-05a)と復習課題(22-05b)、第11回の予習課題(22-11a)と復習課題(22-11b)のそ

れぞれの点数と、2023年度第8回の予習課題(23-08a)と復習課題(23-08b)、第12回の予習課題(23-12a)と復習課題(23-12b)の点数を比較した。受験可能回数が無制限であった2022年度の課題では、ほとんどの学生が満点を取るまで繰り返し受験を行ったが、2023年度では平均点が8~9割で、ばらつきも生じた。予習課題より復習課題の方が、平均点が高くばらつきが少ない傾向も見られた。この差は、受講後にある程度の知識を得たうえで復習課題を受験した学習の効果であると考えられた。

次に、中間試験を廃止し、期末試験のみにしたことが点数におよぼす効果を検討した。2018年度から2023年度までの中間および期末試験の合計点数を比較した結果を図3に示した。2018年度から2022年度までの中間試験、期末試験はそれぞれ100点満点、合計200点満点とし、2023年度は期末試験を200点満点とした。2018年度から2022年度の点数は、図2に示したとおり予習および復習課題で点数差が生じていないため、中間および期末試験の合計点が図1で示した合計点と同様の推移となっている。2020年度から2022年度までかなり高得点側に分布している理由の1つとして、学生間で「過去問」がやり取りされている可能性が考えられる。このことは、実際に何名かの学生が授業者である筆者らに話してくれたことであり、また点数分布の上昇が、教科書とともに試験の設定問を変更した2019年度の翌年度から生じていることも整合的といえる。これに対し、2023年度に実施した期末試験では、平均点が2018年度、2019年度よりは高いが、2020年度から2022年度

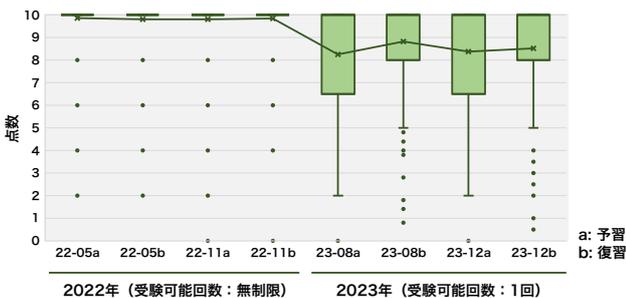


図2 2022年と2023年度の予習・復習課題の点数比較

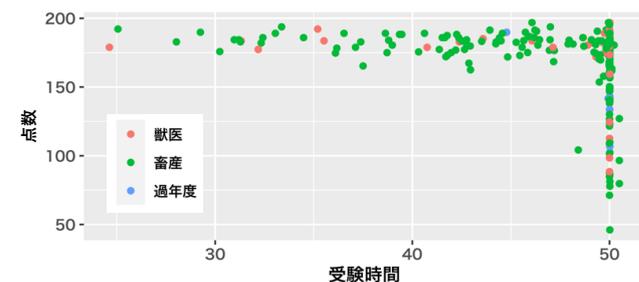


図4 2023年度期末試験の点数と受験時間の散布図

よりは低くなった。

以上より、受験可能回数を1回とすることで、予習および復習課題で分散を広げる高い効果が得られたといえる。また、2023年度の期末試験では問題数を増やし制限時間を短くすることも含めた効果が認められた。以下、2023年度期末試験の受験時間と点数の関連性について検討する。

5. 2023年度期末試験についての考察

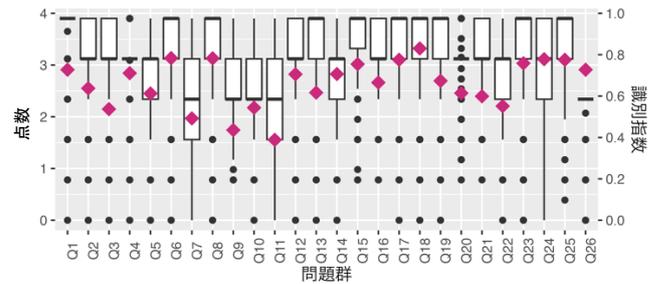
5.1 期末試験の受験時間と点数の関連

2023年度期末試験の問題数および制限時間はこれまでの中間試験（60問30分）と期末試験（60問30分）を足し合わせたものと比べ、問題数はやや増加した128問で制限時間は50分と短縮した。受験時間と点数（スコア）の分布を図4に示した。受験時間は、小テストモジュールの受験結果における「所要時間」として取得した。点数の分布を見ると、受験時間と点数の間に正の相関は認められなかった。獣医、畜産、過年度履修生の点数はそれぞれ概ね同等の分布を示した。最も短い25分で解答した受講生は獣医と畜産に1名ずつおり、点数はいずれも約90%という高得点だった。受験時間が45分程度までの学生のほとんどが75%以上得点しており、受験時間が50分に近づくほど、その受験時間における最低点数が下がる傾向が認められた。受験時間が50分の場合の点数は最高点と最低点の幅が最も大きかった。これらのことから、高得点を取る受講生は制限時間より短時間ですべての解答を終える傾向にあり、得点率の低い受講生は制限時間をすべて活用する傾向にあることが明らかとなった。したがって、制限時間を短くする、または問題数を多くすることは、平均点を下げ点数を分散させる効果があると考えられた。

5.2 期末試験の時間経過に伴う得点の変化

これまでの中間試験や期末試験と同様に、2023年度の期末試験においても、問題は5問ずつ表示した。問題は全128問を通じて受講生ごとにシャッフルされているため、一度に表示される5問もその順序や組み合わせは異なっている。ここではこのような、順序や組み合わせに関わらず、受講生の個々に対して一度に表示される5問（126～128問目では3問）を問題群とした。これに関して、受講生が解答した問題群ごとの点数の分布と識別指数の関係を図5に示した。ここで、識別指数は各問題群の点数と、その問題群の点数を除いた試験合計点数との相関係数であり、これが高い問題群ほど全体の点数をよく説明でき成績の識別性が高いと評価できる。本結果は、小テストモジュールの受験結果を元に生成した。

この図では、Q1からQ26は5問（または3問）の設問からなる問題群で、開始直後に表示されたものがQ1、終了直前に表示されたものがQ26となっている。つまり、図5では時間経過（解答順序）に伴う得点状況の変化を検討する。縦軸の点数はこの試験を100点満点、各問題群の合計点を3.9点満点とした場合の数値を示した。この解析によると、時間の経過に伴う識別指数の変化は概ねランダムで、約0.7の前後で推移していることが明らかとなった。また、得点分布についても顕著な時間変化の傾向は見られなかつ



た。一般に多数の多肢選択問題からなるテストでは、終盤図5 2023年度期末試験の問題群（5問）と点数の関係（黒いグラフは点数、ダイヤモンド型は識別指数を示す）

の問題で当てずっぽうの解答等が増えて徐々に識別指数や得点率が下がったり、逆に好成績の学生のみが終盤の問題に到達できることから時間とともに徐々に識別指数が上がったりする傾向が生じる可能性が考えられる。本解析において識別指数や得点分布の一貫した時間推移が見られなかったことは、制限時間に対する問題量や内容の質、および問題のシャッフル等の形式が、多様な学生の評価として一定程度の妥当性を有したことを示唆する。ただし、このことはあくまで本試験の妥当性を構成（村山 2012）する要素の1つであり、実際に本試験が本科目の目的から見て妥当であるかどうかについては他の指標から検討することが必要である。例えば、予習・復習課題は、期末試験と同様に本科目における生物学の基礎的知識の定着度合いを評価することが期待されるが、期末試験のように「ハイステークス」（Amrein & Berliner 2002）ではなく学生に精神的な負担の少ない形で行われるという点では、各回の学習内容の準備・理解状況をより適切に反映するものと考えられる。一方、期末試験はそれらの総括として、長期記憶により関わる理解状況を反映するものと思われる。したがって、予習・復習課題の総合計点と期末試験評点との相関係数として、期末試験の基準関連妥当性（東京大学教養学部統計学教室 1994）を評価すれば、本科目における生物学の学習成果を異なる側面から整合的に検証することができると考えられる。この相関係数は0.33と計算されることから、やはり本期末試験が一定の妥当性を持つものと評価することが可能であろう。また、本試験の信頼性の指標としてCronbachの α （岡田 2015）を求めると、個別の設問からは0.91、問題群からは0.96となり、いずれも高水準である。本試験の各設問（群）が整合的に科目の学習成果を測るものであることが示唆される。なお、Q7やQ11のように中央値が低下し識別指数が下がっている問題群には、おそらく全体の中に数問存在する難易度の高い問題が比較的高頻度で含まれていた等の可能性も考えられるが、明確な要因は明らかではない。

5.3 2023年度期末試験の解答時間の変化

図6に2023年度期末試験の問題群ごとの解答時間（受験時間のうち、当該問題群のみに充てられた時間）と点数の関係を示した。解答時間は、学生が5問（または3問）ずつ表示される設問ページを遷移する際に、Moodleの標準ログに記録される時刻から推計したものである。128問を50分で解答する場合、各問題群（5問）あたり平均1.95

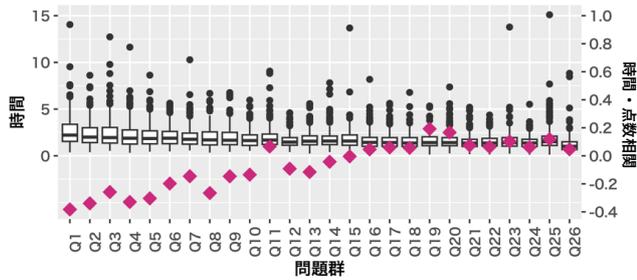


図6 2023年度期末試験の問題群と点数の関係

(黒いグラフは時間、ダイヤ型は各群に要する時間と点数の相関を示す)

分を要することになる。これに対し、多くの受講生は時間経過とともに解答時間が数分の範囲ではあるがやや減少していき、解答時間の幅が減少した。外れ値に相当する受講生についても、試験開始当初は各問題群に対し10～15分

程度の時間をかけた場合もあったが、試験が進行し残り時間が減少するに伴い、試験に費やす時間が減少する傾向にあった。一方で、試験終盤に時間に余裕があった場合は終盤の問題群に時間をかけるケースも認められた。各問題群に費やす時間と得点率(点数)は、Q14までは負の相関関係にあったが、その後は無相関または弱い正の相関となる傾向が認められた。このことは、各問題群に対して時間をかければ高得点が得られるわけではないことを意味しており、むしろ高得点を獲得できる学生は序盤から時間を要せずに解答できるという傾向を定量的に示す結果といえる。したがって、本試験のように時間制約に対して分量の多い多肢選択問題からなる試験において、測定すべき成績は概ね正答数に比例することが示され、この点でも本成績評価の方法が一定の妥当性を有したことを示唆する。

6. 結論

今回、すべての小テストにおいて、問題を変更せず設定を変更することで、成績評価に理想的な点数のばらつきを生じさせることができた。具体的には、講義ごとに行う予習・復習課題の受験可能回数を無制限から1回に変更した。このことが平均点の減少と点数の分散に大きく寄与したことが示された。また、2023年度は中間試験を行わず、期末試験において制限時間を短くし問題数を増加させることで、省力的に平均値を下げ分散を増加させる効果が認められた。期末試験そのものの妥当性は一定程度担保され、信頼性は十分と評価された。全体の効果の確認として、2018年度から2023年度の合計点数を比較した結果を図7に示す。2023年度の総合点では、教科書変更当初の2019年度とほぼ同等に、平均点が減少し、点数の分散も拡大した。図2と図3で示したとおり、この変化に大きく寄与しているのは予習課題・復習課題における変更であったが、試験回数の変更の効果も含まれている。

以上により、Moodleを本実践の方法で利用することにより、大人数の生物学の講義科目において、学習成果を一定の妥当性、信頼性を担保しつつ評価可能であることが明

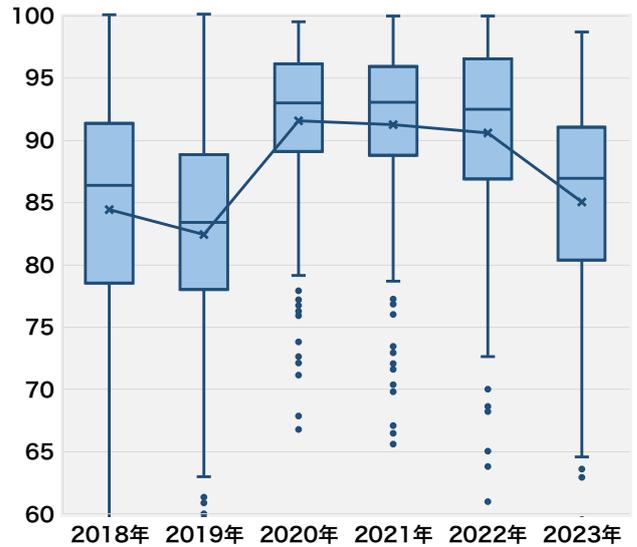


図7 2018～2023年度の総合点の比較

らかとなった。今後は過去問による試験の妥当性、信頼性の低下や、学生が不公平感を抱くこと等を避けるため、期末試験に関する問題バンクのリソースを増やし、学生間で異なる問題セットを出題する等の工夫が必要である。そのための教員負担の増加や、すでに多くの問題を作成した上でさらに作題することの実現可能性も考慮すると、生成AIの活用等、新たな工夫も必要と考える。

なお、例えば予習・復習課題の受験可能回数を1回に制限したことのように、成績評価手法として有効であることと、学習効果をもたらすための教材提供手法として有効であることは、もちろん同じではない。本研究はあくまで前者の有効性を検証したものであるが、後者の観点では、評価課題と学習課題を別にするなどの取り組みが求められる。この点においても、教材のさらなる拡充等、今後の検討が必要である。

謝辞

本論文執筆にあたり、協力いただいた受講生諸氏に御礼申し上げます。

参考文献

- Amrein, A. L., & Berliner, D. C. (2002). High-Stakes Testing & Student Learning. *Education Policy Analysis Archives*, 10, 18-18.
- 有馬 実世 (2021) 「日本における形成的評価概念の受容と今後」人間発達研究, 35, 41-57.
- Davis, B. G. (1993). *Tools for Teaching*. Jossey-Bass. バーバラ・グロス・デイビス (香取 草之助 監訳, 光沢 舜明, 安岡高志, 吉川 政夫 訳 (2002) 『授業の道具箱』 東海大学出版部.
- 中馬いづみ, 得字圭彦, 斉藤 準 (2022). 「課題モジュールのルーブリックによる生物学実験ノートの授業内採点」日本 Moodle 協会全国大会発表論文集, 10, 6-11.
- 中馬いづみ, 得字圭彦, 斉藤 準 (2023). 「課題モジュールのルー

- ブリックを用いた生物学実験の成績評価と検証」日本 Moodle 協会全国大会発表論文集, 11, 11-17.
- 加藤 利康 (2012) 「授業支援システム Moodle における小テストマイニング」『研究報告教育学習支援情報システム (CLE)』8(4), 1-6.
- 村山 航 (2012) 「妥当性概念の歴史的変遷と心理測定の観点からの考察」『教育心理学年報』51, 118-130.
- 野崎 崇弘 & 村川 猛彦 (2019) 「情報リテラシーおよびプログラミングに関する理解度テストの実施と分析」『2019 年度情報処理学会関西支部 支部大会 講演論文集』E-04.
- 岡田 謙介 (2015) 「心理学と心理測定における信頼性について」『教育心理学年報』54, 71-83.
<https://doi.org/10.5926/arepj.54.71>
- 齊藤 準 (2020). 「物理基礎科目における授業外学習の定量的評価」第 9 回大学情報・機関調査研究集会論文集, 68-73
- 齊藤 準 (2021a) 「物理のオンライン授業における H5P, STACK, Essay (auto-grade) の活用」日本 Moodle 協会全国大会発表論文集(9), 28-34.
- 齊藤 準 (2021b) 「物理基礎科目におけるオンライン授業」IDE 現代の高等教育(635), 30-34.
- 東京大学教養学部統計学教室 (1994) 『人文・社会科学の統計学』東京大学出版会.
- 豊田 秀樹 (2000) 『共分散構造分析 [応用編]—構造方程式モデリング—』朝倉書店.
- 和田 武 (2012) 「CMS 小テスト問題分析による授業改善の試み」『学術情報処理研究』16, 167-173.

Moodle に埋め込まれた動的幾何コンテンツに対する JavaScript を用いた学習分析に向けて

齊藤 準

帯広畜産大学

Moodle に埋め込まれた動的幾何ソフトウェアのコンテンツに対し、JavaScript により学習者の学習状況ログを取得する手法について報告する。動的幾何ソフトとしては、GeoGebra と Cinderella を例として使用した。学習状況ログとしては、マウス（タッチ）によるコンテンツ操作の時刻や座標、変数値等を取得した。取得したデータはコンテンツ操作の終了時に、Moodle コース内にあらかじめ用意したデータベースモジュール内へと自動的に保存される実装とした。本実装には齊藤（2023）を援用した。結果として、本研究の手法を用いることで、管理者権限やプラグイン等を要することなく、動的幾何コンテンツに対するラーニングアナリティクスが可能であることが明らかとなった。これによりコンテンツ改善のための情報が得られるだけでなく、成績等の他の学習データからは得られない学習評価への活用が期待される。

Implementation of JavaScript-Based Learning Analytics for Dynamic Geometry Content Embedded in Moodle

JUN SAITO

Obihiro University of Agriculture and Veterinary Medicine

We report on methods to obtain learning logs from Dynamic Geometry Software (DGS) content embedded in Moodle using JavaScript. We utilized GeoGebra and Cinderella as examples of DGS. We obtained the time, coordinates, and values of variables defined upon user interaction with the content, triggered through mouse and/or touch events. We implemented a flow for obtaining data from the interaction and automatically sending and storing it in a database activity module prepared in a Moodle course once the user completes the interaction by utilizing the method developed in Saito (2023). As a result of this implementation, we clarified that it is possible to build a learning analytics (LA) environment capable of obtaining log data from DGS without relying on admin privilege or plugins. It is anticipated that, by applying our methods, it will be possible not only to obtain information to improve DGS content but also to evaluate learning outcomes that are difficult to assess with other learning data, such as usual grading scores.

1. はじめに

動的幾何ソフトウェア（Dynamic Geometry Software、DGS）は、インタラクティブに数学図形やグラフ、分子モデル等を操作し、幾何的証明や解析計算、シミュレーション等を実行可能なコンテンツを作成するためのソフトウェアである。主に STEM（Science, Technology, Engineering, & Mathematics）分野の教材作成に有用である（Kortenkamp 2017, Brehm & Kortenkamp 2004）。

インタラクティブな STEM 分野教材としては、例えばコロラド大学ボルダー校が物理学を中心とする STEM 教育改革プログラムの一環として 2002 年に開発を開始した（Wieman 2017）PhET が知られている。PhET には 160 を超えるシミュレーション教材（University of Colorado Boulder n.d.）が含まれ、現在も新たな教材が開発されている。教材はいずれもブラウザやモバイルアプリで無償利用

できる。PhET のコンテンツと DGS によるコンテンツ（以下、本稿では DGS コンテンツと呼ぶ）の大きな違いは、前者がプロジェクトチームにより作成・提供されるのに対し、後者は授業者自身が自由に、かつ比較的容易な操作で作成できることである。それにより、授業者が個々の授業実践に柔軟かつ最適に準拠した教材を開発できる。

これらのコンテンツのうち、HTML5 と CSS および JavaScript によって構成されるものは、ブラウザのみで使用可能であり、Moodle をはじめとする LMS との親和性が高い。本稿では以下、それらのコンテンツに限定して議論する。なお、関連して、現在は Moodle の標準モジュールにもなっている H5P でも、ブラウザのみで使用可能なインタラクティブ教材を作成できるが、主にインタラクティブなテキストや動画、画像を構成するものであり（H5P Group n.d.）、DGS のような数値・数式計算やシミュレーション教材までには対応していない。

本稿で述べるように、Moodle では、HTML を編集できる場所であればどこでも、DGS のコンテンツを設置できる。その使用場面として、例えば小テストモジュールの中に設置し、コンテンツの操作によって解答を導くような CBT (Computer Based Testing) を開発することが考えられる (安野 2023)。この場合、DGS コンテンツを通じた学習の成果あるいは結果は、CBT への解答として記録され、Moodle の高度な採点・フィードバック機能によって学習者にフィードバックされる。また、小テストの解答時間を通じて、間接的にはあるが DGS コンテンツの使用時間が記録される。このように、DGS コンテンツを Moodle による CBT と組み合わせる場合には、その使用状況が一定程度ログとして残り、また学習者へのフィードバックも可能となる。

一方、DGS コンテンツをそれ以外の場面で、主として学習用教材として設置・使用する場合には、スクリーンキャプチャを取るなどの特殊な方法を用いない限り、コンテンツの使用を通じた学習プロセスの情報は完全に失われてしまう。また、学習者が学習を通じて得た成果とそれに対するフィードバックも、特別な手段を用意しない限り、まったく得られない。しかし、DGS コンテンツの有効性や課題を評価するためには、これらの情報を適切に取得・蓄積し、活用することが必要である。

そこで、DGS コンテンツでの学習状況を詳細にログとして取得・蓄積することが考えられる。ログの取得およびその分析や、それによるフィードバック等の活用は、ラーニングアナリティクス (Learning Analytics, LA, 学習分析) として、さまざまな研究が蓄積されつつある (Ferguson 2012)。

DGS はインタラクティブなコンテンツであり、例えばページやブック、動画といったスタティックなコンテンツのように、単にそれを利用した (読んだ、見た) 時間や回数等から学習のプロセスを得るだけでは十分ではなく、実際に行われた操作の詳細なデータを取得して初めて、プロセス評価が可能となる。JavaScript で構成される DGS コンテンツの場合、JavaScript によってそうした詳細データを取得することができる。

本研究では、DGS に対する LA を Moodle 内で実行するための基礎技術として、Moodle に埋め込まれた DGS コンテンツの操作ログを取得する手法を具体的に実装する。上述の通り、DGS は授業者が独自のコンテンツ・教材を作成できる点に強みがある。したがって、その操作ログの取得も、個々の教材に合わせて授業者自身が実装できるものであることが望ましい。すなわち、管理者権限や、DGS 以外の追加ソフトウェアおよびサービス等を要することなく、通常の教師ロールのユーザである授業者が、LA を実践できる基盤を用意できるものであることが期待される。

以下、本稿ではそのような基盤として、JavaScript によって DGS コンテンツの操作ログを取得し、Moodle 標準のデータベースモジュールへとログを送信・蓄積する具体的方法を示して、その可能性と必要な論点を整理する。なお、同手法は Moodle バージョン 4.3.3 で動作を確認したが、少なくとも本稿執筆時点で Long-Term Support Release (LTS) となっているバージョン 4.1 から最新バージョンである 4.4 までにおいても、Moodle 内コンテンツの HTML 編集による JavaScript コードの埋め込みや、データベースモジュールへのデータの登録といった基本機能は仕様変更されておらず、問題なく適用可能であると考えられる。

2. 対象とする DGS

DGS としては、さまざまなソフトウェア/アプリケーション、サービスが存在する (Bantchev 2010) が、本稿ではそのうち比較的よく用いられている (Yetis & Yarimsakalli 2021, Kortenkamp 2017) GeoGebra (Hohenwarter & Fuchs 2004) と Cinderella (Richter-Gebert & Kortenkamp 1999) に着目する。これらについては、Moodle との連携に関するいくつかの先行研究も知られている (Ferdianová 2017, 亀田&宇田川 2015, Noda, Nakahara & Kaneko 2023a, Noda, Nakahara & Kaneko 2023b, 菰田 2022)。GeoGebra については、数式処理問題タイプである STACK の本稿執筆時点での最新バージョンであるバージョン 4.5.0 に導入されており、オープンソースで開発されているなど、Moodle との親和性も高い。一方、Cinderella はフリーウェアであり、またそのコンテンツを HTML に埋め込むための API である CindyJS がオープンソースで開発されており、やはり Moodle との連携がしやすい。

なお、本稿では GeoGebra や Cinderella と類似の機能を持ち、操作の容易なフリーウェアである Desmos (Empson 2011) についても、比較のため一部の検討に加える。

3. 方法

本研究では、DGS で作成したコンテンツを Moodle 内のページモジュール内に埋め込み、JavaScript によって学習者によるコンテンツの操作ログを取得する手法を実装した。ログの取得・蓄積については齊藤 (2023) の手法を援用し、取得データは、一時的に同じページ内で非表示化 (display hidden) された textarea に書き出し、ブラウザの別ウィンドウにフォーカスが変わったり、ページが閉じられたりしてページのフォーカスが失われるタイミングで、同じ Moodle コース内のデータベースモジュールへと送信・蓄積される仕組みとした。以下の節において、これらの具体的な実装例について述べるが、関連する主な技術仕様の特

徴は表 1 のように整理できる。

表 1 本研究の実装に関連する DGS 別の主な技術仕様

	GeoGebra	Cinderella	Desmos
形態	オープン	フリー	フリー
API	deployggb.js	Cindy.js	Desmos API
iframe	Web/ ファイルから	ファイルから	Web から
Script	JavaScript/ GGBScript	Cindy Script	Desmos API
Script 記述	API/ エディタ	エディタ/ API	API

3.1 DGS コンテンツの埋め込み

本研究で実際に確認した GeoGebra、Cinderella、Desmos の場合、DGS コンテンツを Moodle 内に埋め込むには、大きく分けて、JavaScript API を利用する方法と、Inline Frame (iframe) を利用する方法がある。

前者は JavaScript を使用して、外部のリソースを取得してページ内の JavaScript オブジェクトとして構築する。オブジェクトに対する操作も JavaScript によって行うことができ、JavaScript を用いた LA とも親和性が高い。

一方、後者はさらに、Moodle 外部リソースを Inline Frame 内に埋め込む方法と、Moodle にアップロードされた HTML ファイルを Moodle のファイルモジュールの機能によって Inline Frame 表示する方法に分けることができる。この 1 つ目の場合には、一般に埋め込み側である Moodle と、埋め込まれる DGS コンテンツとは別オリジン（プロトコル、ドメイン、ポート番号）のオブジェクトとなり、ログ取得等、両者の間でのデータのやり取りは、同一オリジンポリシーにより制限される。一方、2 つ目の場合にはこの制限がなく、Moodle と埋め込まれたコンテンツとの間のデータのやり取りが容易になるが、Inline Frame の「外側」領域のコンテンツを自由にデザインできないなどの意味で、使用場面・方法に関する制限はある。

したがって、LA のためのログ取得を念頭に置く場合には、JavaScript API によるコンテンツ埋め込みが簡便・有効である。ただし、ユーザの記述した JavaScript の使用できない Moodle App への埋め込みが必要な場合は、Inline Frame による方法を用いるしかない。DGS コンテンツの利用目的や条件によって、これらの方法を適宜使い分ける必要がある。なお、参考として、GeoGebra のコンテンツをページモジュール内に JavaScript API によって埋め込み

ブラウザで表示した場合の例と、Inline Frame として埋め込み Moodle App で表示した場合の例を、それぞれ図 1、図 2 に示す。

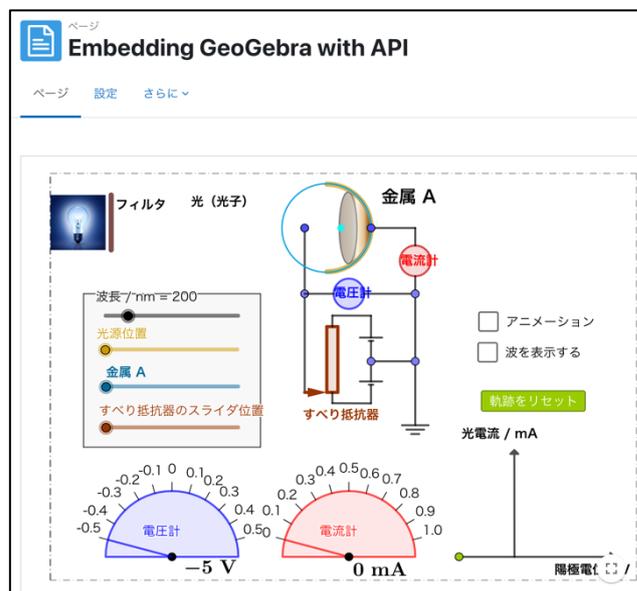


図 1 JavaScript API による DGS コンテンツの埋め込み

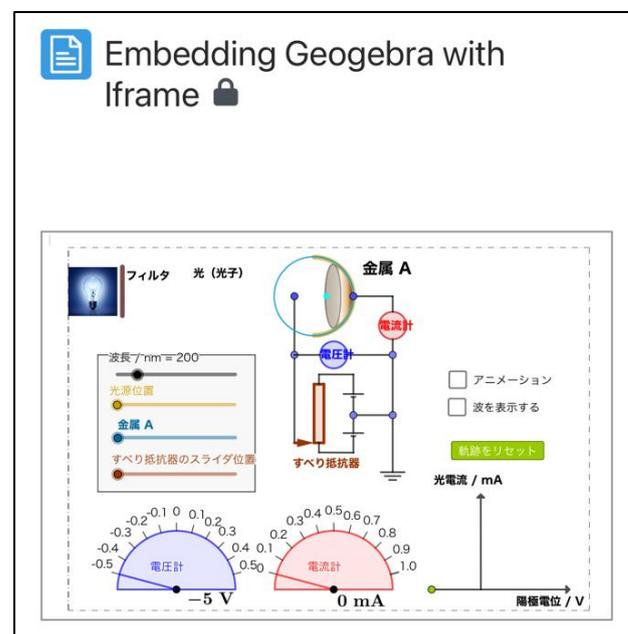


図 2. Inline Frame による DGS コンテンツの埋め込み

本研究では LA への応用可能性を検討するため、JavaScript API の方法によってコンテンツを埋め込んだ。

3.1.1 GeoGebra の場合

GeoGebra で作成したコンテンツは、アカウントがあれば、GeoGebra のウェブサイト内で Materials として保存・公開することができる。保存されたコンテンツに割り振られる ID を参照することで、API により、コンテンツを任意の HTML ページ内に埋め込むことができる。具体的な埋め込みの方法を、一部簡略化（以降も同じ）して以下に示す。

```
<script
src="https://www.geogebra.org/apps/deployggb.js">
</script>

<script>
var params = {
  "material_id": 割り振られる ID,
  appletOnLoad: function(api) {
    // API の操作
  }
};
var applet = new GGBApplet(params, true);

window.addEventListener("load", function() {
  applet.inject('ggb-element');
});
</script>

<div id="ggb-element"></div>
```

3.1.2 Cinderella の場合

Cinderella で作成したコンテンツは、CindyJS という API（ライブラリ）を使って、任意の HTML ページ内に埋め込むことができる。具体的には、まずはコンテンツを HTML でエクスポートする。作成した HTML ファイルの内、script タグで記載されている部分、およびコンテンツ描画要の div タグ（id が CSCampus）の部分のみをコピーし、Moodle のページモジュール等で HTML 編集モードにして貼り付ける。この方法がもっとも柔軟性が高い。

または、作成した HTML ファイルを、ファイルモジュールとしてアップロードする。この場合は、作成したコンテンツが Inline Frame によって Moodle ページ内に表示されることになる。本研究のように、外部のコードとのやり取りをするには、はじめに親ウィンドウを window.parent として取得してから、JavaScript で制御することになる。ここで、Inline Frame のコンテンツはあくまで Moodle にア

ップロードされており、Moodle 側と同一オリジンになっていることが重要である。

本研究ではより簡便な前者の方法をとり、JavaScript によってコンテンツと Moodle 側との情報のやり取りを実装した。具体的なコードを次に示す。

```
<script
src="https://cindyjs.org/dist/v0.8/Cindy.js"></script>

<script id="csmousedrag" type="text/x-cindyscript">
  // ドラッグイベントの Script (CindyScript)
</script>

<script id="csmouseclick" type="text/x-cindyscript">
  // クリックイベントの Script (CindyScript)
</script>

<script id="csdraw" type="text/x-cindyscript">
  // 描画の Script (CindyScript)
</script>

<script type="text/javascript">
  var cdy = CindyJS({
    scripts: "cs*",
    defaultAppearance: {
    },
    geometry: [{ // 描画の記述 }],
    ports: [{width: 1017, height: 524,id: "CSCanvas",
    }]
  });
</script>

<div id="CSCanvas"></div>
```

3.2 JavaScript によるログの取得

DGS から取得可能なイベントとしては、コンテンツ領域内のクリック・タッチ、スライダのドラッグ、オブジェクトの状態変化（フォーカス等）等が考えられる。この他、通常の JavaScript のイベントとして、例えば端末の向きの変化、加速度等、DGS コンテンツで定義されないイベントも扱うことができる。

コンテンツ内のイベントの処理は、API またはコンテンツ作成時点のエディタ内で記述することができる。

3.2.1 GeoGebra の場合

GeoGebra では、GGBScript と呼ばれる独自のスクリプトと、JavaScript が利用できる。前者は、GeoGebra の各種コマンドを操作でき、また内部のイベントに応じてスクリプトをコールできる。ただし、あくまで GeoGebra 内部の制御に用いるものであり、Moodle 側を含め、コンテンツの外部へのアクセスは原則としてできない。本研究で想定するように、ログを取得して Moodle へ送信・蓄積する用途には使用できない。

一方、JavaScript は、GeoGebra コンテンツの埋め込まれたウィンドウの window オブジェクトを取得して、GeoGebra 側から Moodle 側の制御が可能となる。逆に、Moodle 側からは、GeoGebra のオブジェクトを API として取得し、制御することができる。コンテンツ内の操作ログは、コンテンツ内部のイベントリスナーに JavaScript のコードを登録することで可能となる。

または、GeoGebra でコンテンツを作成する時点で、GeoGebra アプリケーション内のエディタで、JavaScript を記述しておくことができる。ただし、この場合は、JavaScript のコードを含めて、GeoGebra のコンテンツが保存されることとなる。コンテンツを GeoGebra のオンラインソースに公開する場合は、コードも公開されることになるため、その内容によってはセキュリティ上の注意が必要である。この問題を避けるには、コンテンツをローカルの HTML としてエクスポートして Moodle にアップロードするか、JavaScript はあくまで API 経由で定義する方法による方が良い。以下はこの後者の方法の具体例である。

```

appletOnLoad: function(api) {
    const disp_la =
        document.querySelector("#disp_la");

    const onMouseDown = function(e) {
        // クリックイベントの処理
        if (e.type === "mousedown") {
            let dt = new Date();
            let coord = "(" + e.x + "," + e.y + ")";
            let log = dt.toLocaleString('ja-JP', {
                timeZone: 'Asia/Tokyo'
            }) + "." + dt.getMilliseconds() + "¥t" + coord +
                "¥n";
            disp_la.append(log);
        }
    };
};

```

```

const onUpdate = function(name) {
    // 変更イベントの処理
    let dt = new Date();
    let v = api.getValue(name);

    let log = dt.toLocaleString('ja-JP', {
        timeZone: 'Asia/Tokyo'
    }) + "." + dt.getMilliseconds() + "¥t" + "voltage¥t" +
        v + "¥n";
    disp_la.append(log);
};

api.registerClientListener(onMouseDown);
api.registerObjectUpdateListener(
    変更対象オブジェクト名, onUpdate);
}

```

3.2.2 Cinderella の場合

Cinderella は、Cindy Script と呼ばれる独自のスクリプトを使用できる。このスクリプトの中に、javascript 関数があり、その引数に JavaScript のコードを記述すると、コンテンツの埋め込まれたウィンドウの window オブジェクトに対する JavaScript を実行できる。コンテンツ内のイベントに対する Cindy Script、および javascript 関数を介した JavaScript は、Cinderella アプリケーションのエディタで記述することができる。

具体的には、上述の Cinderella のコードの内、id が cs で始まる script タグ内に、Cindy Script を記述することになる。ログ取得のためのコード例を以下に示す。

```

<script id="csmousedrag" type="text/x-cindyscript">
    //Script (CindyScript)

    javascript(
        "const disp_la =
            document.querySelector('#disp_la');" +
        "let dt = new Date();" +
        "let v = " + a + ";" +
        "let log = dt.toLocaleString('ja-JP', { timeZone:
        'Asia/Tokyo' }) + '.' + dt.getMilliseconds() + '¥t' +
        v + '¥n';" +
        "disp_la.append(log);"
    );
</script>

```

```
<script id="csmouseclick" type="text/x-cindyscript">
  //Script (CindyScript)

  coord = mouse().xy;

  javascript(
    "const disp_la =
      document.querySelector('#disp_la');" +
    "let dt = new Date();" +
    "let coord = " + coord + ";" +
    "let log = dt.toLocaleString('ja-JP',
      { timeZone: 'Asia/Tokyo' }) + '!' +
    dt.getMilliseconds() + '¥t(' + coord + ')¥n!;" +
    "disp_la.append(log);"
  );
```

3.3 データベースの活用

上述の通り、本研究では各 DGS コンテンツの操作ログを、一度ページ内の textarea に書き出し、これを同じコース内の活動として用意されたデータベースモジュールに送信・蓄積する方法を実装した。データの送信は、ページのフォーカスが外れたタイミングで行われるようにした。以上の手法は、齊藤（2023）で開発されたものであるが、そのコードを一部簡略化して次に示す。

```
const disp_la = document.querySelector("#disp_la");

document.onvisibilitychange = (e) => {
  if (document.visibilityState === "hidden" &&
    disp_la.textContent !== "") {
    let log = disp_la.textContent;
    let data = makeMoodleForm(log);
    let url = "../data/edit.php?id=データベース ID";
    let res = navigator.sendBeacon(url, data);
    disp_la.textContent = "";
  }
};

const makeMoodleForm = (log) => {
  let form = document.createElement("form");
  form.enctype = "multipart/form-data";
  form.action = "../data/edit.php";
  form.method = "post";
  let input = document.createElement("input");
  input.name = "d";
  input.value = データベース番号;
```

```
let input2 = document.createElement("input");
input2.name = "sesskey";
input2.value = getSesskey();
let textarea = document.createElement("textarea");
textarea.name = フィールド ID;
textarea.value = log;
form.append(input, input2, textarea);
let formdata = new FormData(form);
return formdata;
};

const getSesskey = () => {
  let body = document.querySelector("body");
  let pat = /sesskey=([^\s]+)/;
  let sesskey = body.innerHTML.match(pat)[1];
  return sesskey;
};
```

本研究ではパーミッションの設定により、蓄積されたログデータに学生が直接アクセスできないようにすることで、セキュリティ上の課題に対応した。具体的には、まずデータベースモジュールの設定として、エントリーはすべて「要承認」とし、他の学生が登録内容にアクセスできないようにした。これに加えて、パーミッションの mod/data:viewentry から学生を除外することで、ログの登録は可能としつつ、登録されたログは学生からは一切見られないように設定した。ここで、mod/data:view から学生を除外してしまうと、データベースへのアクセスそのものが禁止されるため、ログの登録もできなくなる。なお、より簡便な方法として、データベースモジュールの一覧表示テンプレートや個別表示テンプレートをいずれも白紙にしたり、学生がアクセスしても問題のない情報のみに限定して表示させたりすることも有効であろう。ただしこれらの方法による場合には、教師ロールのユーザもそのままではログを表示できなくなるため、ログの取得にはエントリーのエクスポート等が必要になる。

4. 考察とまとめ

本稿では、Moodle に埋め込まれた DGS コンテンツに対して、その操作ログを取得し蓄積する実装が可能であることを具体的に明らかにした。その際、管理者権限やプラグイン、および外部のソフトウェアやストレージ等は不要であり、通常の教師ロールのユーザがクライアントサイドの JavaScript のみを用いて、マルチモーダルな学習データが取得可能であることを示した。

本研究の方法は、ユーザが記述した JavaScript の動作しない Moodle App では使用できない。したがって、Moodle App も含めた LA 基盤の構築には、バックエンド API の活用等も含めたシステムやプラグイン等の開発が必要となる。この点については、すでに先行研究がある (Noda, Nakahara & Kaneko 2023a, Noda, Nakahara & Kaneko 2023b)。ただし、DGS コンテンツからのログを蓄積するのではなく、あくまでコンテンツ内部で学習者にフィードバックを与えるなどの LA 機能を実装するのであれば、Inline Frame を用いる方法によって、ブラウザと Moodle App に共通する LA コンテンツ化は可能と思われる。例えば、学習者による DGS コンテンツの操作のある種の「実技・実演」として、リアルタイムのログ解析によって一定のパターンに沿った操作を検出・判定しパフォーマンス評価する「実技試験」を行うことなどが考えられる。一般に、LA を含むデータ分析には “Descriptive” “Diagnostic” “Predictive” “Prescriptive” の4種のタイプがあるとされるが (Hernández-de-Menéndez, et al. 2022)、リアルタイムのパフォーマンス評価・フィードバックはそれらの組み合わせによる応用的な手法といえる。具体的な開発実践については、今後検討したい。

マルチモーダルなログデータにより、他の方法では難しい LMS 上の学習プロセスの詳細な評価・可視化が可能となり、教材・教授法の改善、適切なフィードバックの提供等による学生の動機づけの向上等も期待できる。しかし、どのようなプロセス評価やフィードバックが、どのような場面で有効であるかについては、具体的な実践研究により明らかにする必要がある。データの取得・利用には、セキュリティやプライバシーの面で考慮すべきこともある。これらも、今後の課題である。

謝辞

本研究は JSPS 科研費 JP20K03118 の助成および JP22H00085 の支援を受けたものです。

参考文献

- Bantchev, B.B. (2010). A Brief Tour to Dynamic Geometry Software. Retrieved April 25, 2024, from <http://www.math.bas.bg/bantchev/misc/dgs.pdf>
- Brehm, E., & Kortenkamp U. (2004). Advanced Teaching of Geometry with Interactive Tools. (Nagaoka, R., Ishi H., & Hitzer E., Ed.). Proceedings of RIMS-Workshop ITMga 2003.
- Empson, R. (2011). Build And Share Rich Educational Content With Desmos, from <https://techcrunch.com/2011/05/24/build-and-share-rich-educational-content-with-desmos/>
- Ferdiánová, V. (2017). GeoGebra Materials for LMS Moodle

- Focused Monge on Projection. *Electronic Journal of e-Learning*, 15(3), 259-268.
- Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *Int J of Technol. Enhanced Learn.* 4(5/6), 304-317.
- H5P Group. (n.d.). Examples and Downloads | H5P, from <https://h5p.org/content-types-and-applications>
- Hernández-de-Menéndez, M., Morales-Menendez, R., Escobar, C. A., & Ramirez Mendoza, R. A. (2022). Learning analytics: State of the art. *International Journal on Interactive Design and Manufacturing (IJDeM)*, 16(3), 1209-1230.
- 亀田 真澄 & 宇田川 暢 (2015) 「Moodle による数学基礎 e-Learning の取り組み」日本ムードル協会全国大会発表論文集, 3, 11-16.
- Hohenwarter, M., & Fuchs, K. (2004). Combination of dynamic geometry, algebra and calculus in the software system GeoGebra. *Computer algebra systems and dynamic geometry systems in mathematics teaching conference, 2002*, 1-6.
- 菟田 智恵子 (2022) 「Moodle 上の数学グラフ・図形描画機能の拡充」2022 年日本ムードルムート口頭発表.
- Kortenkamp, U. (2017). Developments in Interactive Visualization and Physics Simulation with Cinderella (Study of Mathematical Software and Its Effective Use for Mathematics Education). 数理解析研究所講究録, 2022, 17-28.
- Noda, T., Nakahara, T. & Kaneko, M. (2023a). Monitoring the impact of teacher's intervention in inquiry-based mathematics learning with the use of dynamic geometry. Proc. MathUI2021 – 2022.
- Noda, T., Nakahara, T. & Kaneko, M. (2023b). The learning constructs of inquiry-based mathematics learning represented in the log data of manipulating dynamic content. Proc. MathUI2021 – 2022.
- Richter-Gebert, J., & Kortenkamp, U. (1999). The interactive geometry software Cinderella. Springer.
- 齊藤 準 (2023) 「Moodle 標準モジュールにおける JavaScript の活用」日本ムードル協会全国大会発表論文集, 11, 18-26.
- University of Colorado Boulder. (n.d.). PhET: Free online physics, chemistry, biology, earth science and math simulations. Retrieved April 25, 2024, from <https://phet.colorado.edu/>
- 安野 史子 (2023) 『先生のための CBT 問題事例 & 作成ガイドブック 高等学校 数学・理科』学事出版.
- Yetis, H.S. & Yarimsakalli, M.S. (2021). Teachers' Opinions on Conducting Mathematics Education Online and Recommendations on Improving its Quality. *Int J Edu Sci*, 33(1-3), 74-90.
- Wieman, C. (2017). Improving how universities teach science: Lessons from the science education initiative. Harvard University Press.

フロントエンド API による Moodle と Jupyter 環境の連携

齊藤 準

帯広畜産大学

Moodle と Jupyter 環境とを、フロントエンドの JavaScript のみを用いて簡易連携させる手法について報告する。Jupyter 環境が Moodle とは別のサイト等にある場合、サーバ管理者によって CORS (クロスオリジン間リソース共有) が適切に許可されない限り、フロントエンドのみによる連携は制限される。しかし、本手法では Web Messaging API を使用することにより、学習者側で JavaScript の初期コードを実行する等の操作が一部必要とはなるが、管理者権限や特別なプラグイン、サーバ設定等を必要とせずに、クロスオリジン間であっても Jupyter 環境側のログ取得等の連携が可能となる。報告では、実際の大学でのデータサイエンス教育実践として、国立情報学研究所が運営する JDCat 分析ツールを Binder で無償起動した Jupyter 環境として利用し、学生の学習ログを試行的に取得・分析した結果についても紹介する。これにより、Moodle と Jupyter 環境の連携に関する課題や可能性を議論したい。

Connecting Moodle with the Jupyter Environment Through a Frontend API

JUN SAITO

Obihiro University of Agriculture and Veterinary Medicine

A simple method of connecting Moodle and the Jupyter environment through frontend JavaScript is reported. If the Jupyter environment is located outside the site where Moodle is hosted, connections using only a frontend API, unless CORS (Cross-Origin Resource Sharing) is appropriately allowed, are generally prohibited. However, by using the Web Messaging API, our method enables establishing a cross-origin connection with the Jupyter environment—for example, to extract logs—without any admin privilege, special plugins, or server settings, although it requires students to make some small arrangements, like executing initial JavaScript code. As a particular instance in data science education within an undergraduate curriculum, we report an attempt to obtain learning logs from the Jupyter environment known as the JDCat analysis tool, operated by NII (National Institute of Informatics) and launched using the free-of-charge Binder service. We explore the possibilities and issues related to connecting Moodle with the Jupyter environment.

1. はじめに

データの民主化や、大規模・リアルタイム・マルチモーダルを特徴とするビッグデータ処理の一般化等が進むなか、Industry 4.0 や Society 5.0 といったデータサイエンス・イノベーション関連政策が、国内外を問わず推進されている (内閣府 2016)。そうしたなか、日本では内閣府による AI 戦略がデータサイエンス人材育成の強化を打ち出し (統合イノベーション戦略推進会議 2019)、MDASH 認定制度により大学でのデータサイエンス教育の推進が政策的に後押しされることとなった (数理・データサイエンス・AI 教育プログラム認定制度検討会議 2020)。各大学では、データサイエンス教育強化のため、カリキュラムやコンテンツ等のソフト面の整備とともに、それらを実装するプログラミング環境等のハード・システム面の整備が必要となっている (北川 n.d.)。

OS シェアやデバイスの多様化、および教育機関におけ

る学習者の BYOD (Bring Your Own Device) が進むなか (大学 ICT 推進協議会 ICT 利活用調査部会 2018)、複数の学習者に対して一定共通のプログラミング環境を提供するには、ブラウザのみで動作するクラウドサービスを利用するのが簡便である。そのようなサービスとしては、Google Colaboratory や RStudio Cloud 等、さまざまなものが知られている。とりわけ前者のベースともなっている Jupyter 環境は、クラウド、ローカルのどちらでも利用でき、Python、R、Julia、Octave、Maxima、C++ など、多数の言語に対応していることから (GitHub 2024)、さまざまな形態の教育・学習場面で有効と考えられる。また、Jupyter はオープンソースであり、Moodle との連携活用も期待できる。

Moodle と外部の Jupyter 環境との連携には、さまざまな形態が考えられる。例えば、Moodle の URL モジュール等を使用して、外部の Jupyter 環境へのリンクを設置することも、ごく簡易・表面的な連携の一つとみることが可能であろう。実際、利用者の観点では、個々の外部サービス

のリンクや接続情報を自身で管理することなく、Moodle にアクセスすれば必要な学習材料・環境へとアクセスできる状況になってさえいれば、一定の利便性を果たすものと考えられる。授業者が「手動」で Moodle と Jupyter 環境間のデータを一括エクスポート・インポートすることなども、手軽な連携の一つであろう。

一方で、授業者や管理者の観点からは、さらに踏み込んで、Moodle から Jupyter 環境への教材・資料配布や同期、Jupyter 環境から Moodle への成績・学習状況や提出物の送信・同期・採点、さらにユーザ情報の管理・同期等の機能を含む、より高度な連携が必要となることもあるだろう。実際、配布教材を Jupyter 環境へアップロードしたり、学習結果をダウンロードして Moodle に提出したりする等の作業は、学習者が操作に慣れてさえいれば問題とはならないが、学習者の習熟度やその多様性、使用端末の操作性等によっては、時間を要し必要な学習・授業時間に影響する場合もある。また、授業者にとっては、高度な連携により学習状況の把握やアカウントの管理、課題の採点等の負担が軽減されることは極めて有益であろう。

Moodle と外部サービスとを連携させる強力な方法の一つに、LTI の利用がある。実際、Moodle と Jupyter 環境とを LTI 連携することによるさまざまな先進事例が報告されている(石坂ほか 2019、池田ほか 2022、井関ほか 2022、齊藤ほか 2022、桑田 2022)。あるいは、連携のためのプラグインの活用も考えられ、Moodle と Jupyter 環境とを連携するためのプラグインも開発されている(DNE 2023)。しかしこれらの連携方法は、その導入のために管理者権限やプラグインが利用できる環境等を必要とする。通常の教師ロールである個々の授業者が、「自由に」試すことは難しい。

そこで本研究では、管理者権限や特別な環境的要因に依存することなく、通常の教師ロールの授業者が Moodle と Jupyter 環境とを連携させることのできる方法として、フロントエンド(クライアントサイド)の API である JavaScript のみを用いる実装を検討した。例として、Jupyter 環境でのログを Moodle 側に自動転送し、蓄積する実装を行った。それによる授業の運営、および学習効果の評価を目指した試行を行った。

フロントエンドの JavaScript を用いることの利点は次の通りである(齊藤 2023)。まず、管理者権限や、プラグイン、外部ソフトウェア等が不要である。また、多数のライブラリがウェブ上に公開されており、組み合わせる自由を使用することができる。比較的容易に習得でき、その場ですぐに実装・確認しながら、個々の授業・科目の条件に柔軟・最適に合わせた開発が可能である。

フロントエンド API で外部サービスと連携することは、一般に、CORS の不許可による制限が生じる。本研究では

この困難を回避するため、Web Messaging の仕組みを利用した。これにより、Jupyter 環境のように外部サービス側で JavaScript を使用できる状況では、一般的に Moodle との連携が可能となる。

以下、第 2 章では CORS の制限を回避する連携の方法について詳述し、第 3 章では Jupyter 環境を使用する場合の具体的方法を述べる。第 4 章では本手法による連携の課題と可能性について検討する。

2. CORS と Web Messaging

異なるオリジン(プロトコル、ドメイン、ポート番号)間でのデータのやり取りを、CORS(Cross-Origin Resource Sharing)という(MDN n.d.)。セキュリティ確保のための同一オリジンポリシーにより、接続先のサーバが明示的に許可する設定になっていなければ、フロントエンド API による CORS は一般に禁止される。この制限の例外が、WebSocket API のメッセージング(Web Messaging)である。

Web Messaging は、ブラウザのウィンドウ間でメッセージをやり取りする仕組みである。これにより、テキストを含むさまざまな JavaScript オブジェクトを CORS としてやり取りできる。あくまでウィンドウ間のやり取りであるから、接続のためには、接続元から接続先を別ウィンドウ(タブを含む)で開き、接続が確立した後も両ウィンドウを開いたままにしておく必要がある。また、双方のウィンドウでメッセージのやり取りのための JavaScript が動作する必要がある。

2.1 メッセージの送信

メッセージの送信は、送信先(ターゲット、以下 T と呼ぶ)の window(正確には windowProxy) オブジェクトのメソッドである postMessage を使う。送信先 T の window オブジェクトは、送信元(センダ、以下 S と呼ぶ)の window オブジェクトのメソッドである open で開く。

例えば、“Hello!” というメッセージを、<https://example.target.com/target> というページに送信したい場合、送信元 S の JavaScript コードは次のようになる。postMessage メソッドの第 1 引数にメッセージ、第 2 引数に送信先 T のオリジン(<https://example.target.com>)を指定する。

```

window_target = window.open(
    "https://example.target.com/target", "targetname");
window_target.postMessage(
    "Hello!", "https://example.target.com");

```

2.2 メッセージの受信

このように、メッセージの送信自体は送信元からいわば一方的に実行することが可能であるが、送信先（受信側）がそのメッセージを受け入れるかどうかは、送信先が処理する。具体的には、メッセージが送信されると、送信先で message イベントが発生する。明示的に message イベントに対するイベントリスナーが記述されていなかったり、記述されていてもメッセージを受け取る処理になっていなかったりする場合には、メッセージは単に無視される。これにより、異なるオリジン（クロスオリジン）間で勝手にメッセージを送信されることによる攻撃を防ぐことができる。

例えば、送信先 T で次のような JavaScript コードを記述することにより、メッセージを処理することができる。ここで、event.origin はメッセージ送信元のオリジンを表し、これが想定している送信元 S のオリジン（ここでは https://example.origin.com とする）と一致しているときのみその後の処理を行うよう記述することで、悪意のある攻撃を避けることができる。送信されたメッセージの内容は event.data で取得できる。

```

window.addEventListener("message", (event) => {
  if (event.origin
    !== "https://example.origin.com") return;
  // do something
});

```

2.3 メッセージの返信

このようにして、メッセージの送信と受信が可能になるが、メッセージの「やり取り」には、送信先 T から送信元 S への「返信」も必要であろう。実際には、返信は単に逆方向の「送信」と同じであり、それには postMessage メソッドを使用できる。ここで、postMessage メソッドに必要な返信先 S の window オブジェクトは、返信元 T の window オブジェクトに対する opener として参照できる。これは、返信元 T は、返信先 S によって開かれたウィンドウだからである。

具体的には、例えば返信元 T から返信先 S への“Thanks!”というメッセージの送信は、返信元 T の JavaScript コードとして、次のように記述できる。

```

window.opener.postMessage("Thanks!",
  "https://example.origin.com");

```

なお、メッセージを受信したイベントリスナーの中で、

即座に返信をする場合、返信先の window オブジェクトは event.source としても参照できる。例えば、次のような記述で、受信と返信を同時に処理することが可能である。

```

window.addEventListener("message", (event) => {
  if (event.origin
    !== "https://example.origin.com") return;
  event.source.postMessage("Thanks!", event.origin);
});

```

3. Jupyter 環境と JavaScript

前章では、クロスオリジン間のメッセージ送受信の方法を示した。本章では、クライアントサイドの JavaScript を用いた Moodle と Jupyter 環境との接続例として、Jupyter 環境の操作ログを取得して Moodle へ送信し、Moodle 内に蓄積する実装を示す。これにより、フロントエンド API のみを用いて Moodle と外部サービスとのクロスオリジン間連携が可能であることを、具体的に明らかにする。

3.1 JDCat 分析ツール

本研究では、クラウドの Jupyter 環境として、日本学術振興会 (JSPS) が開発し、現在は国立情報学研究所 (NII) が運用している「JDCat 分析ツール」(日本学術振興会 n.d.) を利用した。このツールは GitHub のリポジトリから Binder によって構築した Jupyter 環境を提供するものであり、本稿執筆時点では、“ac.jp”または“go.jp”のメールアドレスを持っていれば、誰でも無償で使用することができる。構築される Jupyter 環境は、各ユーザに 1 台あたり 5GB (以前は 3GB) のメモリ、10GB の保存容量が割り当てられ、最大 10 台まで構築できる。環境とその中の保存データは、最後の使用から 30 日間は保存される。大学等の演習や研修、研究プロジェクト等での利用が想定されている。

環境構築のベースとなる GitHub リポジトリに置いたファイルを、構築された Jupyter 環境のホームディレクトリにそのまま配置することができる。これを利用すれば、教材配布も容易に可能である。ただし、GitHub リポジトリは一般公開する必要があるため、公開に向かない教材の配布には別の手段が必要であろう。

本ツールでは、ユーザー一人一人に個別の Jupyter 環境が構築され、その URL もランダムに異なる。したがって、通常の外部サービスのように、LTI による連携や、単に Moodle から構築環境へリンクを張る方法による連携は困難である。

3.2 データベースモジュールの活用

そこで本研究では、まずユーザに一人一人異なる自分の構築環境の URL を取得してもらい、それを Moodle 側のコース内に用意したデータベースモジュールに登録してもらった。データベースは「要承認」の設定とし、Moodle 上の学生ロールのユーザが自分で登録した URL のみを参照できるようにした。この URL をクリックすれば、Moodle 内から、構築環境へと飛ぶことができるようになる。図 1 は、実際に学生ユーザが登録したエントリをデータベースモジュール内で一覧表示した画面のスクリーンショットである。ユーザの氏名や URL に含まれるユーザ名の部分は加工してある。



図 1 データベースに登録されたエントリ（氏名等加工）

ただし、学生が毎回データベースに入って URL にアクセスするのは煩雑と考えられる。本実践では、データベースに登録した URL へジャンプするボタン（ランチャ）を、Moodle 内に JavaScript で表示させることとした。実装したコード（一部簡略化）は次の通りである。学生が当該ページにアクセスすると、fetch がこの学生の権限でデータベースへアクセスする。データベースのテンプレート設定で、登録した URL は、"theurl" という name の a（アンカー）タグとして一覧表示されるようにしてあり、fetch で取得されたページから、正規表現を使ってこのタグの href プロパティ（すなわち URL）を読み出すようにしている。学生が複数の URL を登録していた場合は、ページ内の最後に表示されるもの、すなわち最新の URL が取得される。button 要素を新たに生成し、click イベントで、この URL を window.open で開くようにしている。これで、このボタンがランチャとなって、外部 Jupyter 環境を起動することができるようになる。ここで、window.open で Jupyter 環境側の window オブジェクトへの参照が取得され、前章で述べたように、これを用いて Web Messaging によるやり取りも可能となる。

```
const db =
  "https://example.moodle.com/mod/data/view.php?
  id=モジュール ID";
```

```
const lbl = "Launch JupyterLab";
const name = "theurl";
const re = new RegExp("<[^\>]*" + name + "[^\>]*>[^\<]*
  <a[^\>]*href=.(.+).[^\>]*>[^\<]*</a>", "g");
fetch(db)
  .then((response) => response.text())
  .then((text) => {
    const mats = [...text.matchAll(re)];
    if (mats.length == 0) return;
    const thelink = mats[mats.length - 1][1];
    if (typeof thelink !== "undefined") {
      const launch =
        document.querySelector("#launch");
      const btn =
        document.createElement("button");
      btn.textContent = lbl;
      btn.addEventListener("click", (e) => {
        window_open =
          window.open(thelink, "jupyterlab");
        return false;
      });
      launch.append(btn);
    }
  });
```

齊藤（2023）に詳述されているように、Moodle では教師ロールのユーザが HTML を書き込める場所であればどこでも、JavaScript を書くことができる。本実践では、ページモジュール内にスクリプトを記述し、そこから Jupyter 環境を起動できるようにした。図 2 は、ページモジュールにおいて、コンテンツに上記の JavaScript コードを記述した編集画面のスクリーンショットである。ドメインやモジュール ID は加工してある。また図 3 は、このページを実際に表示させた画面のスクリーンショットである。

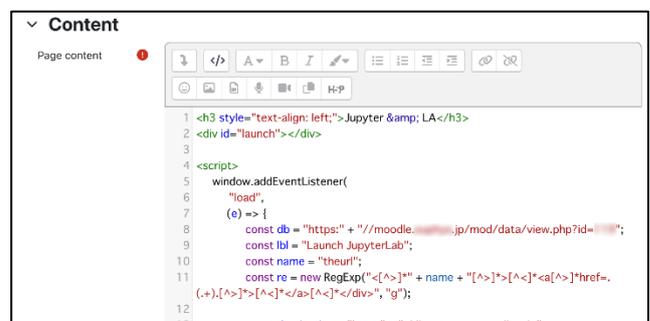


図 2 Moodle 内での JavaScript の記述例（一部加工）

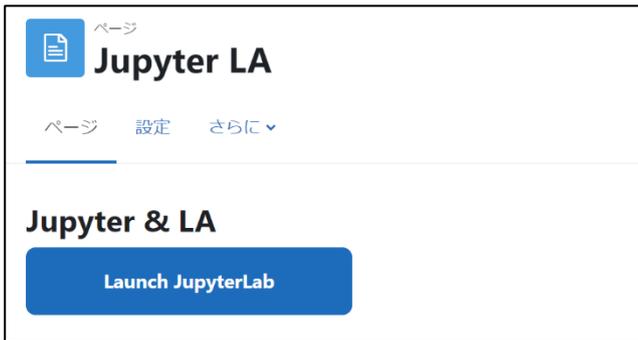


図3 JavaScriptで生成された起動ボタンを含むページ

なお、もちろんブラウザのブックマーク等に URL を保存しておけば、各ユーザは構築環境へと容易にアクセスできるのであるが、大学等の授業科目で使用する場合、自分の PC 以外の端末から作業するなどの理由で、ブックマーク登録等の方法が使用できない場合もある。何より、ウィンドウ間のメッセージングを使用する本研究の手法で Moodle と外部 Jupyter 環境とを連携接続させるには、ブックマーク等によることは難しい。

3.3 Jupyter 側での JavaScript の実行

本研究では、Jupyter 環境からのログ取得を主な連携の目的としていることから、Moodle 側から Jupyter 側へのメッセージ送信は特に活用しなかった。ログ取得には、Jupyter 側から Moodle 側へのメッセージ送信と、Moodle 側での受信・処理が必要となる。この前者のためには、Jupyter 側で JavaScript を実行する必要がある。本研究は Jupyter 環境で Python と R による演習を行うデータサイエンス科目を具体的な実践対象として想定しており、どちらの言語の場合も、Jupyter のカーネルの機能によって JavaScript を実行できる。以下、その具体的方法を述べる。

3.3.1 メッセージ送信のためのイベント

本研究では、Jupyter 環境内での学習状況ログを取得するため、Jupyter 環境内の Notebook セルを実行する度に、実行したコードとその出力内容（エラーも含む）を Moodle 側へとメッセージ送信する設計を行った。Jupyter の Notebook は、セルの実行を含めてユーザが操作（インタラクション）を行うと、セルのフォーカスやスタイル（デザイン）が切り替わる。HTML のレベルでは、それらは HTML 要素の class 属性の変化として表現される。このような HTML 文書の変化は、MutationObserver インターフェースによって「監視」でき、文書の変化に応じて postMessage を実行するようすれば、ユーザによる Jupyter 環境の操作ログ（Notebook での学習状況）を取得することができる。コードの具体例は以下に示す。

3.3.2 Python の場合

Jupyter 環境内の Python は、IPython カーネルによって

動作する。このカーネルの機能の一つとして、ユーザがセル内に記述した JavaScript を、当該ウィンドウ内で実行させることができる。具体的には、マジックコマンド %%js を記入したセルに JavaScript のコードを入力し、通常のセルと同様に実行することで、JavaScript を使用することができる。セルを実行しない場合、コードが記載されていても JavaScript が実行されないことに注意を要する。

Jupyter 環境の Notebook では、セルを実行すると、その次のセルにフォーカス（選択）が移り、フォーカスの移ったセルの class 属性は jp-mod-selected に切り替わる。この class 属性の変化を MutationObserver により監視し、属性の変化したセルの直前にあるセルを兄弟要素（previousSibling）として取得すれば、ユーザが実行したセルやその出力を参照できる。これを postMessage によって、Jupyter 環境のオープナーである Moodle 側のページ内へと送信することで、ログ取得・送信が可能となる。一部簡略化したコードの例を示す。

```

%%js
let dupl = false;
const target = document.querySelector(".jp-Notebook");
const config = { attributes: true,
  attributeFilter: [ "class" ] };
const callback = (mutationList, observer) => {
  const origin = "https://example.moodle.com ";
  for (const mutation of mutationList) {
    if (!dupl) {
      if (mutation.type === "attributes") {
        const current = mutation.target
          .querySelector(".jp-mod-selected");
        const prev = current.previousSibling;
        if (window.opener) {
          window.opener.postMessage(
            prev.innerText, origin);
        }
      }
    }
    dupl = !dupl;
  }
};
const observer = new MutationObserver(callback);
observer.observe(target, config);

```

なお、実際に変更されたセルの内容を取得すると、同じ内容が二重になることがわかった。これは、同じセルに対して class の変更が 2 度行われるためと推測されるが、上

記のコードでは複製状況を変数 `dupl` で取得することで、複製が送信されないようにしている。図 4 は、実際の Jupyter 環境内のセルに JavaScript コードを入力した様子のスクリーンショットである。ドメインは加工してある。

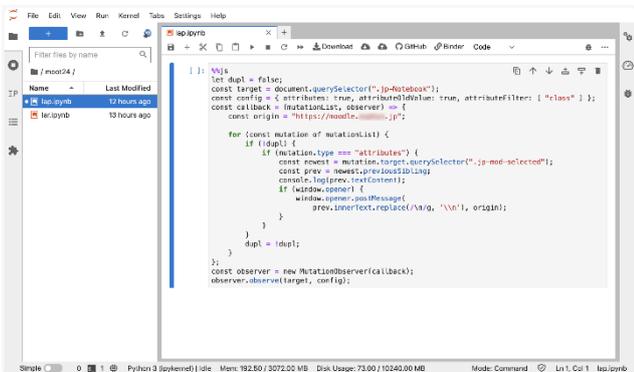


図 4 JavaScript コードが入力されたセル（一部加工）

3.3.3 R の場合

R の場合には、IRkernel に組み込まれている IRdisplay パッケージで、`display_javascript` 関数を利用することで、任意の JavaScript コードを記述・実行できる。

```
IRdisplay::display_javascript('
// ここに Python の場合と同じ JavaScript コードを記述
')
```

図 4 の Python の場合と同様に、R の場合も Jupyter 環境内のセルに上記のようなコードを入力して実行する。図 5 は実際のスクリーンショット（ドメインは加工）である。

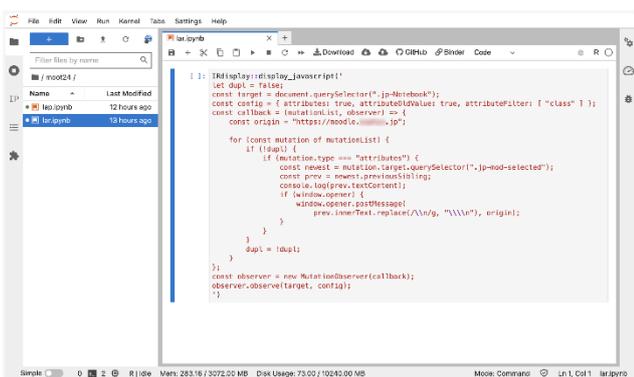


図 5 JavaScript コードが入力されたセル（一部加工）

3.4 Moodle 側でのメッセージ処理

以上によって、Jupyter 側の操作ログを Moodle に送信することが可能となる。Moodle 側では、送信されたメッセージをメッセージイベントで受信・処理できる。

本研究では、齊藤（2023）で示された手法を使用し、送

信されてきたメッセージを同じコース内のデータベースモジュールの中に蓄積する実装とした。これにより、通常は管理者権限の必要となる別途データベースサーバの運用や連携を要することなく、教師ロールのユーザが管理可能な Moodle コース内の環境だけを用いて、データの蓄積が可能となる。

具体的には、まずメッセージイベントで、受信したメッセージをページ内で `hidden` とした `textarea` 内に書き込む処理とした。次にこのページのフォーカスが外れたタイミング（例えば、タブを切り替えたり、ブラウザを閉じたりするなど）を `visibilitychange` イベントとして取得し、Navigator インターフェースの `sendBeacon` メソッドを使用して `textarea` 内のテキストをデータベースへと送信するとともにクリアする設計とした。データベースへの登録はあくまでそのユーザの権限・情報で行われる。これは、ユーザが開いているページ内に含まれるセッションキーを取得することで実現できる。以下、一部簡略化したコード例を示す。

```
const disp_la = document.querySelector("#disp_la");
const destination = "https://example.jupyter.com";
window.addEventListener("message",
  (event) => {
    if (event.origin !== destination) return;
    let dt = new Date();
    let log = dt.toLocaleString('ja-JP', {timeZone:
      'Asia/Tokyo'}) + "%t" + event.data + "%n";
    disp_la.append(log);
  });
document.onvisibilitychange = (e) => {
  if (document.visibilityState === "hidden" &&
    disp_la.textContent !== "") {
    let log = disp_la.textContent;
    let data = makeMoodleForm(log);
    let url = "../data/edit.php?id=データベース ID";
    let res = navigator.sendBeacon(url, data);
    disp_la.textContent = "";
  }
};
```

```
const makeMoodleForm = (log) => {
  let form = document.createElement("form");
  form.enctype = "multipart/form-data";
  form.action = "../data/edit.php";
  form.method = "post";
  let input = document.createElement("input");
```

```

input.name = "d";
input.value = データベース番号;
let input2 = document.createElement("input");
input2.name = "sesskey";
input2.value = getSesskey();
let textarea = document.createElement("textarea");
textarea.name = フィールド ID;
textarea.value = log;
form.append(input, input2, textarea);
let formData = new FormData(form);
return formData;
};
const getSesskey = () => {
  let body = document.querySelector("body");
  let pat = /sesskey=(["\']+)$/;
  let sesskey = body.innerHTML.match(pat)[1];
  return sesskey;
};

```

なお、Jupyter 環境からのメッセージは、あくまで Moodle 側の Jupyter 起動ページで受信するため、このウィンドウを閉じてしまうと、メッセージ受信が成立しない。フォーカスは不要であるが、ユーザはウィンドウを開いたまま Jupyter 側の操作を行う必要がある。最終的にはこのウィンドウを閉じる（フォーカスを外す）ことで、データベースへとデータが送信される。

3.5 実際の例

本章でここまで述べてきた Jupyter 環境と Moodle との連携動作について、以下、簡単な実例を示す。

図 6 は、Python の動作する Jupyter 環境側で 3.3.2 項の JavaScript コードと、`print("Hello, world")` という Python コードを実行した様子のスクリーンショットである。

```

[1]: %js
let dupl = false;
const target = document.querySelector(".jp-Notebook");
const config = { attributes: true, attributeOldValue: true, attributeFilter: [ "class" ] };
const callback = (mutationList, observer) => {
  const origin = "https://moodle.localhost.jp";
  for (const mutation of mutationList) {
    if (!dupl) {
      if (mutation.type === "attributes") {
        const newest = mutation.target.querySelector(".jp-mod-selected");
        const prev = newest.previousSibling;
        console.log(prev.textContent);
        if (window.opener) {
          window.opener.postMessage(
            prev.innerHTML.replace(/&#x/g, '\\n'), origin);
        }
      }
      dupl = !dupl;
    }
  }
};
const observer = new MutationObserver(callback);
observer.observe(target, config);

[3]: print("Hello, world")
Hello, world

```

図 6 Python コードの実行例（一部加工）

図 6 では、この Python コードが正しく実行され、画面に Hello, world という文字列が表示された様子が示されている。実は、`printt("Hello, world")` という誤った（t が 1 つ多い）コードが実行されてエラーが出力された後に、ユーザが正しいコードに上書き修正してから実行し直したのが同図なのであるが、そうした経緯をこの結果からうかがい知ることはできない。

図 7 は、図 6 のコードの実行状況がそのまま Moodle のページに送信されている様子のスクリーンショットである。ここでは可視化のためページにログが表示されるようにしているが、実際には非表示とすることが可能である。

```

Jupyter & LA
Launch JupyterLab

2024/7/11 18:14:54.825 [1]:%js\nlet dupl = false;\nconst target = document.querySelector(".jp-Notebook");\nconst config = { attributes: true, attributeOldValue: true, attributeFilter: [ "class" ] };\nconst callback = (mutationList, observer) => {\n  const origin = "https://moodle.localhost.jp";\n  for (const mutation of mutationList) {\n    if (!dupl) {\n      if (mutation.type === "attributes") {\n        const newest = mutation.target.querySelector(".jp-mod-selected");\n        const prev = newest.previousSibling;\n        console.log(prev.textContent);\n        if (window.opener) {\n          window.opener.postMessage(\n            prev.innerHTML.replace(/&#x/g, '\\n'), origin);\n        }\n      }\n      dupl = !dupl;\n    }\n  }\n};\nconst observer = new MutationObserver(callback);\nobserver.observe(target, config);

```

図 7 Moodle に送信されたログの例（一部加工）

さらに、図 8 は、別のタブのページが表示されるなどでこのページのフォーカスが外れ、ログがデータベースへと自動的に送信された様子を示す。各セルの実行された時刻やその入力内容、エラーを含めた出力内容がすべてデータベースのエントリとして記録されていることがわかる。

```

2024年 07月 11日
Last modified: 2024年 07月 11日

log
pittle.Moodle.JS: Jupyter LA |
2024/7/11 18:14:54.825 [1]:%js\nlet dupl = false;\nconst target = document.querySelector(".jp-Notebook");\nconst config = { attributes: true, attributeOldValue: true, attributeFilter: [ "class" ] };\nconst callback = (mutationList, observer) => {\n  const origin = "https://moodle.localhost.jp";\n  for (const mutation of mutationList) {\n    if (!dupl) {\n      if (mutation.type === "attributes") {\n        const newest = mutation.target.querySelector(".jp-mod-selected");\n        const prev = newest.previousSibling;\n        console.log(prev.textContent);\n        if (window.opener) {\n          window.opener.postMessage(\n            prev.innerHTML.replace(/&#x/g, '\\n'), origin);\n        }\n      }\n      dupl = !dupl;\n    }\n  }\n};\nconst observer = new MutationObserver(callback);\nobserver.observe(target, config);
2024/7/11 18:15:16.448 [*]:\nprint("Hello, world")
2024/7/11 18:15:16.674 [2]:\nprint("Hello, world")
-----\nNameError: name 'printt' is not defined
2024/7/11 18:15:29.417 [3]:%js\nlet dupl = false;\nconst target = document.querySelector(".jp-Notebook");\nconst config = { attributes: true, attributeOldValue: true, attributeFilter: [ "class" ] };\nconst callback = (mutationList, observer) => {\n  const origin = "https://moodle.localhost.jp";\n  for (const mutation of mutationList) {\n    if (!dupl) {\n      if (mutation.type === "attributes") {\n        const newest = mutation.target.querySelector(".jp-mod-selected");\n        const prev = newest.previousSibling;\n        console.log(prev.textContent);\n        if (window.opener) {\n          window.opener.postMessage(\n            prev.innerHTML.replace(/&#x/g, '\\n'), origin);\n        }\n      }\n      dupl = !dupl;\n    }\n  }\n};\nconst observer = new MutationObserver(callback);\nobserver.observe(target, config);
2024/7/11 18:16:02.625 [*]:\nprint("Hello, world")
2024/7/11 18:16:02.667 [3]:\nprint("Hello, world")\nHello, world

```

図 8 データベースに蓄積されたログ（一部加工）

一般に、プログラミング学習における途中のエラーやその上書き修正の有無を、学習者以外が把握することは難しい。また、エラーや修正等がなかったとしても、学習者がコードのどの部分にどの程度の時間を要したのか、ある時

点で学習者がどこまでコードを作成しているのか、といった情報も、教員等が完成したコードから確認することはやはり困難である。しかし、ここで見たように、本章で述べた方法によれば、こうした学習プロセスをログとして詳細かつリアルタイムに取得・評価することが可能となる。

3.6 授業での試行

以上の実装を、実際の授業の中で使用し、Jupyter 環境内からの学習状況ログの取得を試行した。試行は地方国立 O 大学の 2 年次生を対象とする情報処理演習科目において行った。履修者数は 22 名であった。

3.3 に示した JavaScript コードをあらかじめ最初のセルに記載した Python 用および R 用の Notebook ファイルを用意し、これを使用して当該授業の演習を行うよう学生に依頼した。上述の通り、学生は忘れずに最初のセルを実行する必要があり、また、Jupyter 環境を起動した Moodle 側ページも開いたままにしておく必要がある。

結果として、Jupyter からのログがうまくデータベースに記録された学生は、わずか 2 名にとどまった。データの取得に失敗した原因としては、学生が Notebook 先頭の JavaScript のコードを実行しなかったこと、Jupyter の起動ページを閉じてしまったこと、用意した Notebook ファイルを使用しなかったこと、等が推察された。いずれにせよ、本研究の方法でログを適切に取得するためには、その実施方法について学生への十分な説明が必要であることが示唆された。

4. 考察と議論

4.1 本研究のまとめ

以上により、本研究の方法によって、クロスオリジン間であっても、Moodle の通常の教師ロールのユーザがフロントエンドの JavaScript のみを用いて、Moodle と Jupyter 環境とを連携し、ログデータの取得が可能であることを具体的に明らかにした。その際、CORS の制限は、Web Messaging によって回避可能であり、併せて Jupyter 環境としては Binder で構築されたものでも問題ないことを明らかにした。また、ログの蓄積のためには、齊藤 (2023) の手法により Moodle コース内に設置したデータベースモジュールのみで良いことを確認した。これにより、LTI 連携や特別なプラグイン、管理者権限等を要することなく、原理的にさまざまな外部サービスと Moodle を連携できる可能性が明らかとなったといえる。

本研究では、テキストベースでのログ取得を行ったが、メッセージ送受信で実装可能な範囲内で、他の連携方法も可能であろう。例えば、Moodle 側からの教材配信や、Jupyter 側からの課題提出等も、技術的には可能と思われる。

これらについては、今後具体的な実装を検討したいと考えている。

4.2 本研究の限界

本研究では、教師ロールのユーザが、外部の Jupyter 環境からログ取得が可能であることを具体的に明らかにしたが、実際の授業等での本格的なログの取得およびその分析は未着手である。本研究の方法では、Jupyter Notebook で入力・実行された内容がそのままテキストで取得されるため、実際の分析にはしかるべきテキスト処理等が必要となる。その具体的方法や、それにより得られる情報の有効性等については、今後明らかにする必要がある。

本研究は管理者権限等が不要な実装を目指すため、フロントエンド API のみでの連携を行ったが、連携先の外部サービスで JavaScript を自由に記述・実行できるとは限らない。したがって、本研究の手法が適用できるのは、一部の外部サービス・システムに限定されることも留保が必要であろう。

また、第 3 章に述べた通り、ユーザが Jupyter 環境側で JavaScript のコードを実行することや、メッセージ送受信をするウィンドウ間の接続を維持することなど、本研究の手法に必要な条件が実際の授業実践等では担保されない可能性があることにも留意が必要である。これらへの対応も今後の課題としたい。

謝辞

本研究は JSPS 科研費 JP20K03118 の助成を受けたものです。

参考文献

- 大学 ICT 推進協議会 ICT 利活用調査部会 (2018) 『BYOD を活用した教育改善に関する調査研究結果報告書 (第 1 版)』.
- DNE - Ministère de l'Education Nationale. (2023). Moodle plugin directory: Jupyter Notebook. Retrieved November 4, 2023, from https://moodle.org/plugins/mod_jupyternotebook
- GitHub. (2024). Jupyter kernels. Retrieved April 25, 2024, from <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>
- 池田 裕希・長尾 和彦・田房 友典 (2022) 「Moodle と JupyterHub を用いたプログラミング環境の構築」『情報処理学会第 84 回全国大会講演論文集』2022(1), 205-206.
- 井関 文一・浦野 真典 (2022) 「LTI カスタムパラメータによる Moodle - JupyterHub 連携に関する研究」 MoodleMoot Japan 2022 Proceedings.
- 石坂 徹・桑田 喜隆・合田 憲人・政谷 好伸・横山 重俊・浜元 信州 (2019) 「Moodle と Jupyter Notebook の連携によるプログラミング教育環境の構築」日本ムードル協会全国大会発表論文集, 7, 32-37.
- 北川 源四郎 (n.d.) 「コンソーシアムの歩み (2017~2021 年度)」 Retrieved April 25, 2024, from <http://www.mi.u->

tokyo.ac.jp/consortium/pdf/first_activitiy.pdf

- 桑田 喜隆 (2022) 「CoursewareHub を用いたプログラミング教育支援」 AXIES 年次大会企画セッション (2022 年) .
- MDN (n.d.) 「オリジン間リソース共有 (CORS)」 Retrieved April 25, 2024, from <https://developer.mozilla.org/ja/docs/Web/HTTP/CORS>
- 内閣府 (2016) 『第 5 期科学技術基本計画』 Retrieved April 25, 2024, from <https://www8.cao.go.jp/cstp/kihonkeikaku/5honbun.pdf>
- 日本学術振興会 (n.d.) 「データの検索・分析 (JDCat)」 Retrieved April 25, 2024, from <https://www.jsps.go.jp/j-di/search.html>
- 齊藤 準 (2023) 「Moodle 標準モジュールにおける JavaScript の活用」 日本ムードル協会全国大会発表論文集, 11, 18-26.
- 齊藤 智也・王 躍・西井 淳・末長 宏康・大平 康旦・西村 世志人・金山 知余・村永聡・爲末 隆弘・岡田 耕一・レール マルク (2022) 「Moodle と JupyterHub を用いた Web 型プログラミング教育環境の改善事例: 複数科目・複数講師による Nbgrader の共同利用」 大学 ICT 推進協議会 2022 年度年次大会.
- 数理・データサイエンス・AI 教育プログラム認定制度検討会議 (2020) 『「数理・データサイエンス・AI 教育プログラム認定制度 (リテラシーレベル)」 の創設について』 Retrieved April 25, 2024, from <https://www8.cao.go.jp/cstp/ai/suuri/ninteisousetu.pdf>
- 統合イノベーション戦略推進会議 (2019) 『AI 戦略 2019～人・産業・地域・政府全てに AI～』 Retrieved April 25, 2024, from <https://www8.cao.go.jp/cstp/ai/aistratagy2019.pdf>

査読なし論文 / Non-refereed Papers

学生の省察支援ツールとしての Moodle 活用 ～活動モジュール比較検討～

浅田義和^{†1} 村岡千種^{†2}

^{†1}自治医科大学 ^{†2}藤田医科大学

高等教育において、学習者の省察は重要な要素の1つである。この省察を支援するためのツールとして、ミニッツペーパーや大福帳などが存在する。本研究では、Moodle を省察支援ツールとして利用するにあたり、どのような活動モジュールが利用可能かを比較検討した。学習者が省察を記載・提出し、個別のフィードバック等を行わない場合であれば小テストやアンケートが利用可能である。記載内容に対してコメントや評価などを加える場合は、フォーラムやデータベースなどの機能を活用する必要がある。従来は紙媒体であったミニッツペーパー等を電子化することで様々な利点も生じるが、その特徴をどのように活用し、どのように教育に役立てていくかによって、活動モジュールの使い分けを入念に検討する必要がある。

Moodle Usage as a Tool for Reflection Support - Comparative review of activity modules -

YOSHIKAZU ASADA^{†1} CHIKUSA MURAOKA^{†2}

^{†1}Jichi Medical University ^{†2}Fujita Health University

Learner reflection is one of the key elements in higher education. As tools to support the reflection, tools such as minute papers and Daifuku-Cho exist. In this study, a comparison was made as to what kind of activity modules are available for using Moodle as a reflection support tool. Quizzes and questionnaires can be used if the learner writes and submits a reflection, and no individual feedback is given. If comments and evaluations are to be added to the written content, it will be necessary to utilize functions such as forums and databases. Although the digitization of minute papers, etc., which used to be paper-based media, offers various advantages, it is necessary to carefully consider how the different activity modules should be used, depending on how the features are to be utilized and how they are to be used for education.

1. はじめに

高等教育において、学習者の省察は教育上で重要な要素の一つである。学習者の振り返り・省察を促すためのツールとして、ミニッツペーパーや大福帳などが存在する(中井 2015)。ミニッツペーパーは授業の終了時に学生にコメントを書かせるものであり、学生の理解度確認にも利用することが可能である。大福帳は学生との双方向性を強化したものであり、教員がコメント等を返せるような仕組みとして運用するものである。これ以外にもジャーナル、コメントシート、など多様なものが存在している。本論文では、以上をまとめて「省察支援ツール」と呼ぶ。

従来、省察支援ツールはミニッツ「ペーパー」や大福「帳」という名前で見られるように、紙媒体を基本として使われることが多かった。Moodle をはじめとした LMS の導入が進んだことなどの理由から、電子媒体としての省察支援ツールが利用されるケースも登場してきている(早川 2017, 牧野 2021, 田中 2022, 中園 2023 など)。

オンラインで省察支援ツールを利用する方法は多数存在している。一方で、何らかの専用システムを利用することになる以上、その管理運営の手間も生じることとなる。例えば田中(2022)の例は Google Form を主として利用しているが、学内で別途 LMS を利用する場合などは管理の手間が二重になってしまう欠点も生じる。また、早川(2017)の例は JavaScript を利用しており、ソースコードの管理・メンテナンスの必要性が生じる可能性もある。

本論文では、LMS の1つである Moodle を利用し、その機能で省察支援ツールがどのように運用可能であるか、比較検討したことを報告する。

2. 方法

Moodle 4.3 を用いて、その活動の中から省察支援ツールとして利用可能なものを整理し、著者間での意見交換・合意を行った。なお、Moodle には多数のプラグインが存在するが、今回は一部のプラグインのみを比較検討の対象とし、主に標準機能として搭載されている活動モジュールに

焦点を当てることとした。対象の活動モジュールは以下のとおりである。なお、()内の表記はモジュール名称であり、*のついたモジュールは追加プラグインである。

- フォーラム (forum)
- 小テスト (quiz)
- 課題 (assignment)
- データベース (database)
- 用語集 (glossary)
- フィードバック (feedback)
- * アンケート (questionnaire)
- * 日誌 (journal)
- * Diary (diary)

また、検討にあたっては、省察支援ツールの利用に際して、以下の観点を確認することとした。末尾の2文字は結果を整理する際の略称である。

- 学習者本人への影響が大きいもの
 - テキスト以外での入力形式：形式
 - 入力後の内容確認や修正：確認
 - 複数回の入力対応：複数
 - 教員からのコメント機能：コメ
 - コメント等の際の通知：通知
- 同級生や教員など、他者からみた機能
 - 学生同士での相互閲覧：相互
 - 採点・評価：採点
- 特に教員側の評価や学習分析で有用な機能
 - 内容の集計：集計
 - 一括ダウンロード：DL

以下、上記の活動モジュールについて個別に整理する。

3. 結果

3.1 フォーラム

フォーラムでは学習の振り返り等を記載するにあたり、テキストが利用可能である。ラジオボタン等での選択式の回答を入力することはできない。学習者が入力した内容は、投稿後も再確認が可能であり、一定時間以内であれば修正も可能となる。フォーラムの設定であえて限定した場合を除いて、複数件の入力も可能である。投稿内容については返信が可能であり、メール購読している状態であれば通知も送信可能である。

学習者同士での閲覧に関しては、デフォルトではフォーラム投稿内容は教師・学生問わず全員が可能になっている。しかし、分離グループの機能を使い、学生1人ずつを異なるグループに割り振った状態で利用することで、他の学生には見えない状況を作り出すことも可能である。なお、この分離グループ作成については、1グループあたりの人数を1人と指定することで、Moodleのグループ作成機能から自動で実施可能である。採点機能をONとすることで教

師は投稿内容の採点が可能である。パーミッションの設定を修正することで学生同士での評価も可能になる。

テキスト情報が主体となるため、自動で集計するような機能は有していない。フォーラム投稿内容のエクスポートについては、CSV形式等で出力が可能となっている。

なお、これはフォーラムに限ることではないが、評価に関して、絵文字を用いた尺度を作成して設定することで、5つ星での評価やFacebook等にみられる「いいね」の評価を擬似的に再現することもできる。投稿内容をCSV・Excel等の形式でダウンロードすることは可能である。

3.2 小テスト

小テストは多肢選択やCloze形式などを利用することで、テキスト入力以外の内容も利用可能である。複数回の回答は可能であるが、一度提出してしまった内容を修正・上書きすることはできない。作文問題を用いた場合は、テンプレートや記載例をプレースホルダとして提示しておくことも可能である。それ以外の形式でも設問の本文等で例を示すことはできる。教員からのコメントは作文問題であれば可能であるが、基本的には事前に指定した内容のフィードバックとなる。採点時の通知等はない。

また、内容閲覧や採点は教師に限定されている。小テストの標準機能で統計情報を確認することも可能であり、結果をCSV等でエクスポートすることも対応している。

3.3 課題

課題はオンラインテキストやファイル提出を実施することができるが、ファイル提出の場合は一括ダウンロードなどの作業が煩雑になることも踏まえ、ここでは「オンラインテキスト」に限ったものとして記載する。課題提出については提出完了処理をしない限りは内容の確認や修正も可能である。複数件の投稿を行うには、1回毎に再オープンするなどの対応をするか、活動モジュールを複製するか、という対応が必要になる。教員からのコメントを記載することができ、評価後の通知を行う機能も有する。

小テスト同様、基本的には学習者を評価するための機能として設定されており、採点や閲覧は教師権限でのみ可能となる。内容の集計については採点結果を確認できる程度であり、一括ダウンロードは1学生1ファイルとなる。

3.4 データベース

データベースは初期設定にこそ時間が必要となるが、テキストやラジオボタンなど、様々な形式を組み合わせた入力フォームを設定することができる。データベースのプリセットとして登録されている「日誌」や「提案」を基本とすることで設定の負荷を減らすことも可能となる。入力後の内容確認や修正についても基本的に可能であり、複数件の投稿にも対応している。コメントをつけることも可能であるが、フォーラム等と異なり、コメントに対する自動通知機能は有していない。

学生同士での閲覧については「投稿を承認」することで可能となる。また、採点については承認されたものに対してのみ、実施できる。フォーラム同様、採点機能はデフォルトで教師権限のみであるが、パーミッションを設定することで相互評価も可能となる。

内容を自動集計する機能は存在していないが、入力内容を一括ダウンロードすることができるため、この結果を用いて分析することは可能となる。

3.5 用語集

用語集は簡易版のデータベースとも言える機能である。入力欄が語句と定義の2箇所となる以外は、データベースと同様、エントリの投稿や承認、採点、出力などが可能となっている。このため、「タイトルと投稿内容」といった2点のみの投稿であれば用語集でも実施可能であり、「タイトル、投稿内容、5段階の自己評価」など複数の入力項目を利用する場合は用語集では不可能となる。

3.6 フィードバック

フィードバックは主としてアンケートを実施するための機能である。自由記述以外にもラジオボタンでの選択などを作成することが可能である。複数回の入力については、匿名であれば回答可能である。実名の場合は前の回答を上書きする設定となってしまうため、毎日の投稿を促す場合などは注意が必要となる。

フィードバックは前述の通り、実名かつ複数回入力となっている場合、前の入力内容を確認して上書きができる。それ以外の場合には、回答者自身では何を入力したか、確認する術がない。また、個別のコメントや採点をする機能も有していない。

回答内容の集計は自動で実施され、設定次第では学習者にも提示することが可能である。また、入力結果の一覧もダウンロードできる。

3.7 アンケート*

アンケートは標準機能ではなく、Moodleのプラグインサイトで公開されている機能の1つである。フィードバックと同様に、学習者に対するアンケート実施を主たる目的として作成されたものとなる。

フィードバックとの比較として、設問形式が豊富になっている点のほか、匿名かつ複数回の登録ができる点、提出後の内容が自分で確認できる点などが特徴である。

一方、基本的な位置づけとしてアンケートであることには差異がないため、投稿された内容に対する採点や返信コメントなどを行う機能は有していない。内容の集計や集計結果の公開・共有、一括ダウンロード等についてはフィードバックと同等である。

3.8 日誌*

日誌もアンケート同様、標準機能以外のプラグインとして存在するものである。日誌は1つの活動モジュールに対

して1人1件の投稿が可能であり、その形式はテキストに限られる。投稿した内容の確認・編集は可能である。

教員からのコメント機能は有しており、フィードバック実施時には課題等と同様に学習者への通知が実施される。学生同士での日誌閲覧の権限は有しておらず、集計やダウンロードに関する機能も存在しないが、採点についてはフォーラム等と同様、評価の設定を行うことで実施できる。

3.9 Diary*

Diaryは日誌機能の改良プラグインとして登録されているものである。1つの活動モジュールに複数の投稿ができる点、字数制限などを細かく実施できる点などの改良点が存在する。なお、本稿執筆時点での最新版(3.7.9)では、日本語対応がなされていない。

考察

本研究では、6つの標準モジュールと3つの追加プラグイン、合計9つの活動モジュールについて検討を行った。結果欄ではそれぞれの特徴を個別に整理したが、表1・表2として、省察支援ツールの検討に用いた観点別にまとめたうえで考察を加える。

△をつけた点は以下の理由からである。

- 小テストのコメント返信は作文問題に限定される
- フィードバックの複数回答は匿名に限定される
- 小テストの採点では尺度が利用できない
- 小テストの集計は設問単位となる
- 課題のダウンロードは1人1ファイルとなる

表1・表2とも、すべてが○となる活動モジュールは存在していなかった。そこで、省察支援ツールの利用場面を想定しつつ、考察を加えることとする。

表1 学習者の立場から

活動モジュール	形式	確認	複数	コメ	通知
フォーラム	X	○	○	○	○
小テスト	○	○	○	△	X
課題	X	○	X	○	○
データベース	○	○	○	○	X
用語集	X	○	○	○	X
フィードバック	○	X	△	X	X
アンケート*	○	○	○	X	X
日誌*	X	○	X	○	○
Diary*	X	○	○	○	○

表2 他者の立場から

活動モジュール	相互	採点	集計	DL
フォーラム	○	○	X	○

小テスト	X	△	△	○
課題	X	○	X	△
データベース	○	○	X	○
用語集	○	○	X	○
フィードバック	X	X	○	○
アンケート*	X	X	○	○
日誌*	X	○	X	X
Diary*	X	○	X	○

表1は主に学習者にとってどのような支援が可能かという観点である。そこで、その支援の方向性ごとに整理する。

形式：ラジオボタンやチェックボックスでの回答を可能とするか否かという観点である。文章での記述を促すだけでなく、理解度を5段階評価させるなどの設問を加える場合には有用な項目である。このため、初学者にとってはテキスト以外の形式も設定できる方が利便性は高いと考えられるが、ある程度省察に慣れた学習者を対象とするのであれば、この項目はXであっても良いと考えられる。

確認：省察をあとから確認することは学習の重要な要素であり、この項目がXとなってしまうフィードバックは注意が必要である。逆にこれ以外の機能であれば、学習者は過去の入力内容を振り返ることが可能である。

複数：入力回数については、仮にXとなっていた場合でも、活動モジュールを複製して利用することで対応は可能である。しかし、例えば30日間の省察を毎日入力する場合に30個のモジュールが存在することは非常に煩雑である。学習者の利便性からは、○であることが必要であろう。

コメ：教員からのコメントについては、省察支援ツールの利用方法に大きく関わる。ミニツペーパーとして意見を集めた後、次の授業等で教員がその内容を整理したうえで紹介・フィードバック等を行うのであれば、個々人に返信ができなかったとしても省察の支援には有用である。むしろ、1人ずつ個別に返信できてしまう場合、学部・領域によっては1教員で100人以上に対してコメントを返すという可能性もあり、人的な負荷を避ける意味ではXとなっていることにも意義はある。一方で、特に学習支援の必要な学習者などの特例に対しては、何らかの形で個別にフィードバックを行うことも意義がある。この場合はXの活動モジュールについては別途でメッセージ等を用いたコミュニケーションをとることが求められる。

通知：仮にMoodle上からの通知が自動でなされなかった場合でも個々人に直接「返信をした」ことを伝えることは可能であり、必要性はやや低いものである。

表2では学習者本人以外、すなわち教師または学習分析等を行う立場での観点となる。

相互：相互閲覧については、利用する場面次第となる。

学習者同士でのコミュニケーション、協働学習などを意図する場合は有効活用できる。一方で、他人に見えてしまうことで十分な省察ができないという学習者の存在も考慮すると、場面に応じて使い分ける必要があるだろう。

採点：省察支援ツールとして使う場合、10点満点などの素点での採点よりも、3段階・5段階などの尺度を利用することの方が多くと考えられる。このため、小テストのように点数でのみ採点されるものはやや利便性が低くなってしまいうことも考えられる。一方、特に小テストについては表1の観点では有用なものであり、場面に応じて使い方を検討する必要があるといえる。

集計：集計機能は小テスト・フィードバック・アンケートでのみ利用可能なものであるが、授業中に全体的な傾向を示すような活用が可能となる。このため、前述のように個別フィードバックが困難な場合の次善策として考えることができるであろう。

DL：結果のダウンロードについてはほぼすべての活動で実施可能である。特に複数の活動モジュールを比較検討する際は有用となる。集計の機能と合わせ、個別の学習・教育の機会には不要な場合もあるが、省察支援ツールの効果を検証する際などには有用と考えられる。

以上の考察を踏まえて、改めて活動モジュール別の特徴を整理する。小テストやアンケートは個別のコメントを不要とする場合は有用である。これはある程度の省察が自分でできる学生に対しての支援ツールの使い方となる。一方、省察が不慣れである学生については、個別のフィードバックや指導が必要となる。この場合はフォーラムやデータベースを用いた個別返信を含めた指導が有用となろう。

今回、フィードバックモジュールは「確認」がXであり、記載した本人が後から内容を振り返ることができないというデメリットがあった。一方、標準機能のみで考えた場合は「集計」が可能という特徴がある。大学全体でMoodleを運用している場合など、プラグインの追加が容易にできない場合などは、利用を検討する意義もあるだろう。

本研究の限界として、省察支援ツールの定義がやや漠然としている点が挙げられる。本研究ではミニツペーパーや大福帳などを起点としたものを対象としたが、従来、紙媒体で利用されていたものを電子化したことで、様々な利点が追加されていることも考えられる。実際、ICTを活用した教育の改善に際して、Puentedura (2003) によるSAMRモデルやKimmons et al.(2020)によるPICRATモデルなどが提唱されており、ICT活用以前からの教育を代替するのか、あるいは強化・転換するのか、といった観点で検討する必要性も述べられている。省察支援ツールの特徴を整理するにあたっては通知機能やダウンロードなどICT独自の視点も追加しているが、この点は少し入念に整理する必要もあるだろう。

2 点目に、2 名の著者のみによる合意である点が挙げられる。MoodleMoot Japan 2024 での発表も行ったものの、改善に関する意見などをより広く集めることも必要であろう。この際は、教師としての意見のみならず、学生としての意見、学習分析や教学 IR などデータを集約して扱う立場の意見なども広く集める必要があろう。

最後に、プラグインについては広く調査しておらず、標準機能を主体としたことも挙げられる。Moodle のバージョンアップにともない、標準機能以外で利便性の高いものがあっても継続的に利用できるか否かの判断を行いつづらうこともある。今回はこのような理由からプラグインの調査は最低限としたが、中園（2023）のように日誌（journal）プラグインを利用した研究実践報告なども存在しており、ある程度の範囲では調査も必要と考えられる。

結論

学習者の省察支援ツールとして Moodle を用いるにあたり、どの活動モジュールが利用可能かを検討した。各学習者の省察内容を収集することに重きを置くのであれば小テストやアンケートの利便性が高い。個別フィードバックを行うのであればフォーラムやデータベースが有用である。一方、電子化されたことで省察支援ツールにも様々な特徴が生じうるため、その利点・欠点を全体として整理し直すことなども今後の課題として明らかとなった。

参考文献

- 早川美徳（2017）．オンライン版「大福帳」を用いた授業改善．大学 ICT 推進協議会 2017 年度 年次大会論文集．
https://axies.jp/_files/report/publications/papers/papers2017/WA1-5.pdf, Accessed April 25, 2024.
- Kimmons, Royce, Graham, Charles R., West, Richard E. (2020). The PICRAT model for technology integration in teacher preparation. *Contemporary Issues in Technology and Teacher Education*, 20, 1, 176-198.
- 牧野みのり, 尾澤重知 (2021)．リアルタイム型授業における振り返りと学習成果の関連．日本教育工学会研究報告集 JSET2021-1, 180-187
- 中井俊樹（2015）．アクティブラーニング．玉川大学出版部, 202.
- 中園長新（2023）．教員と学生の相互交流を実現する「大福帳」型オンラインミニツツペーパーの運用と課題．麗澤大学紀要. 106, 40-51
- Puentedura, Ruben R (2003). A matrix model for designing and assessing network-enhanced courses. <http://www.hippasus.com/resources/matrixmodel/index.html>, Accessed April 25, 2024

田中雅章（2022）．ミニレポート提出による学習習慣の確立．第 17 回医療系 e ラーニング全国交流会 講演要旨集. 38-39