

Moodle 用動画配信システムにおける視聴履歴の可視化

渡邊 康太[†] 藤本 茂雄^{††} 檜垣 泰彦^{†††}

[†]千葉大学大学院融合理工学府 〒263-8522 千葉市稲毛区弥生町 1-33

^{††}千葉大学国際未来教育基幹 〒263-8522 千葉市稲毛区弥生町 1-33

^{†††}千葉大学アカデミック・リンク・センター 〒263-8522 千葉市稲毛区弥生町 1-33

E-mail: ^{†††}higaki.yasuhiko@faculty.chiba-u.jp

あらまし 千葉大学では、メディア授業を目的として、SCORM パッケージによる Moodle 上での動画配信システムの開発を行った。しかし現行のシステムでは、学生の視聴行動を詳細に分析することが難しい。そこで、学生の視聴行動を折れ線グラフを用い表現することで、今後のオンデマンド授業の改善や、教材の開発に有益なデータを提供することができるツールの開発を行った。グラフを作成する際に必要な視聴ログを保存する方法として、データ数を抑えつつも学生の視聴行動を正確に把握する事ができる操作イベント方式が有効な手法であることを示した。また getElementByClassName 方式を用いることで、データベースへの安全なアクセスとデータの取得が可能となる。視聴ログを折れ線グラフに変換し、教師が学生の視聴行動を視覚的に理解することができるツールの開発が可能であることを確認した。

キーワード Moodle, 動画配信システム, 視聴ログ, 可視化, メディア授業, SCORM パッケージ

Visualization of Viewing Logs in Moodle-based Video Delivery Systems

Kota WATANABE[†] Shigeo FUJIMOTO^{††} Yasuhiko HIGAKI^{††}

Graduate School of Science and Engineering, Chiba University 1-33 Yayoi-cho, Inage-ku, Chiba-shi, 263-8522 Japan

^{††}Institute for Excellence in Educational Innovation, Chiba University 1-33 Yayoi-cho, Inage-ku, Chiba-shi, 263-8522 Japan

^{†††}Chiba University Academic Link Center 1-33 Yayoi-cho, Inage-ku, Chiba-shi, 263-8522 Japan

E-mail: [†]higaki.yasuhiko@faculty.chiba-u.jp

Abstract At Chiba University, development of a Moodle-based video delivery system using SCORM packages was conducted for the purpose of media instruction. However, the current system faces challenges in analyzing students' viewing behaviors in detail. To address this issue, a tool was developed to represent students' viewing behaviors using line graphs, aiming to provide valuable data for improving on-demand instruction and developing educational materials. The effectiveness of the operation event method in accurately capturing students' viewing behaviors while minimizing data volume required for graph creation was demonstrated. Additionally, the use of the getElementByClassName method enables secure database access and data retrieval. It was confirmed that converting viewing logs into line graphs allows for the development of tools that facilitate teachers' visual understanding of students' viewing behaviors.

Keywords Moodle, Video Delivery Systems, Viewing Logs, Visualization, Media-Based Courses, SCORM Packages

1. はじめに

千葉大学では 2020 年度からの全員留学に備え、メディア授業を目的として、SCORM パッケージによる Moodle 上での動画配信システムの開発を行った[1]。このシステムでは教師側が学生の視聴状況（スコア＝動画の何パーセントを視聴済みか）を LMS API 機能により管理している。現状のシステムでは主に学生の視聴管理に焦点を当てており、実際に学生がどのように

授業動画を視聴しているのかは分からない。

そこで本研究では学生の動画視聴の様子をログデータとして残して、教師側が学生の視聴状況を視覚的に確認できるツールの開発を行う。これにより学生の視聴行動を分析することができ、新しいビデオ教材を作る際の参考データが得られる可能性がある。

本システムで開発するツールは、図 1 のように縦軸を「動画の再生位置」、横軸を「動画を再生してからの

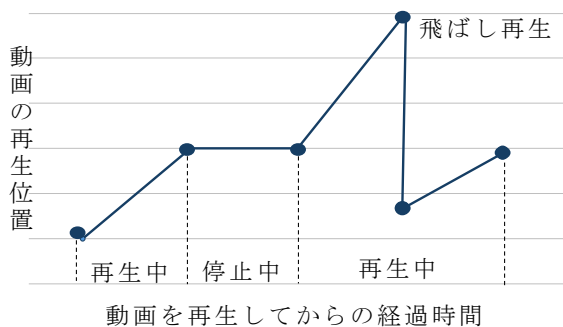


図 1 可視化グラフのイメージ

表 1 フィールド名と保存するデータ

フィールド名	保存するデータ
UserID	視聴者のユーザーID
EntryTime	プレイヤーを起動した時刻
CurrentTime	動画の再生位置
ElapsedTime	動画を再生してからの経過時刻

経過時間」の折れ線グラフ(以下可視化グラフとする)とする[2]。このツールを用いて教師は、動画の一時停止されている箇所や飛ばされている箇所を視覚的に把握することで、学生の視聴行動を分析することができる。

2. 視聴履歴

Moodle を用いたオンデマンド授業の利点として、学生は時間と場所を選ばずに何回でも授業動画を視聴できる点が挙げられる。そのため本システムでは、コースを受講している多数の学生が複数回、授業動画を視聴する場合を想定し、可視化グラフを作成するために必要なデータベースモジュールのフィールドを表 1 のように 4 つ定義した。

UserID フィールドには Moodle の「アカウント名」が保存され、これによって個々の学生を一意的に識別することが可能となる。EntryTime フィールドには各学生が「当該動画にアクセスした時刻」を保存する。これらの 2 つのデータを組み合わせることで、どの学生がいつ動画を視聴したかを判断することができる。また CurrentTime フィールドには可視化グラフの縦軸の値に該当する「動画の再生位置」が保存され、ElapsedTime フィールドには可視化グラフの横軸の値に該当する「動画を再生してからの経過時間」が保存される。

Moodle コースでの操作ログをコース内に設置したデータベースモジュールに送信、蓄積する方法が提案されている[3]。

ここでは Moodle API 機能を用いて、上記 4 つのデー

表 2 学生の操作イベント

操作イベント	意味
Play	再生
Stop	一時停止
RateChange	倍速変更
Seeking	シーク操作開始
Seeked	シーク操作終了
Exit	ページ離脱

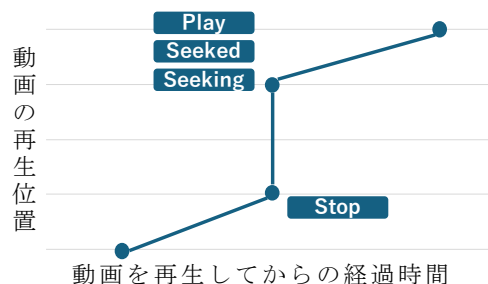


図 2 動作を再生しながらシーク操作を行った場合の可視化グラフ

タ(以下視聴ログとする)を Moodle のデータベースモジュールに保存する。その際、視聴ログを保存するタイミングとして操作イベント方式とインターバル方式の 2 つの手法がある。

2.1. 操作イベント方式

操作イベント方式では、表 2 に示す操作イベント(「Play」「Stop」「Seeking」「Seeked」「Exit」「RateChange」)が発生したタイミングで視聴ログをデータベースに保存する。

この手法では、学生が動画プレイヤーに対して操作を行ったタイミングで視聴ログが保存されるため、データ量を最小限に抑えながら可視化グラフを作成することができる。例えば、学生が動画を再生してから 1 回も動画プレイヤーに対して操作を行わずページを離れた場合、操作イベントとして「Start」と「Exit」の 2 回のみであり、2 つの視聴ログで可視化グラフを作成することができる。

またこの手法では、「動画を再生しながらシーク操作を行った場合」と「動画を停止した状態でシーク操作を行った場合」で区別して、保存する視聴ログの値を変更する必要がある。

まず動画を再生しながらシーク操作を行った場合、図 2 に示すようにシーク操作の開始から終了までの間に「Stop」「Seeking」「Start」「Seeked」の 4 つの操作イベントのタイミングで視聴ログが保存される。その際、「Stop」のタイミングで取得する「CurrentTime」は、シーク操作を開始した時点での動画の再生位置となり、

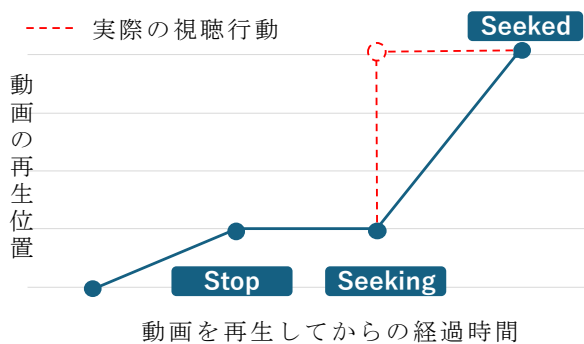


図3 動画を停止しながらシーク操作を行った場合の可視化グラフ

「Seeking」「Start」「Seeked」のタイミングで取得する「CurrentTime」はシーク操作が終了した時点での動画の再生位置となる。

次に動画を停止した状態でシーク操作を行った場合、図3に示すようにシーク操作の開始から終了までの間に「Seeking」と「Seeked」の2つの操作イベントのタイミングで視聴ログが保存される。その際に取得される「CurrentTime」は、「Seeking」と「Seeked」共にシーク操作が終了した時点での動画の再生位置となり、実際の学生の視聴行動とは異なるグラフが作成されてしまう。そのため動画を停止した状態でシーク操作を行う場合は「Seeking」のタイミングで取得される「CurrentTime」をシーク操作が開始した時点での動画の再生位置に変更する必要がある。そこでまず、「Timeupdate 関数」を用いて、常に動画の再生位置を取得し、事前に作成した配列の中に保存する。これにより常に現在の動画の再生位置とひとつ前の動画の再生位置を配列に保持した状態でシーク操作を行うことができ、「seeking」のタイミングで、配列から一つ前の動画の再生位置を視聴ログとして保存する事で、実際の学生の視聴行動を再現した可視化グラフを作成することができる。

2.2. インターバル方式

インターバル方式は、一定の間隔で視聴ログを保存する。この手法では、操作イベントが発生したかどうかに関わらず視聴ログを一定の間隔で保存し続けるため、操作イベント方式のような複雑な処理が不要である。また、操作イベントが発生していない場合においても視聴ログが保存され続けるため、データ数が非常に多くなってしまふ。例えば、1時間の動画において5秒間隔で視聴ログを保存する場合を考える。その場合、1回の視聴当たりに保存される視聴ログは720個となり、多くの学生が視聴する講義では、膨大なデータ数になることが想定される。しかしデータ数が多い分、正確な可視化グラフを作成することができるという利点がある。

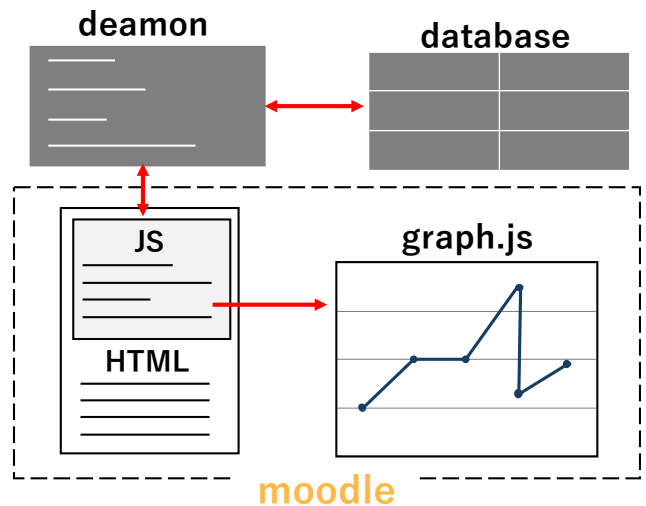


図4 デーモン方式の概要

3. DBに保存されたデータの利用

可視化グラフの作成には、データベースモジュールに保存した視聴ログを用いる。その際、視聴ログの取得方法としてデーモン方式と `getElementByClassName` 方式の2つの手法が考えられる。

3.1. デーモン方式

デーモン方式は図4に示すように、MoodleとMoodleのデータベースサーバー間でデータをリアルタイムにやり取りし、動的なグラフ描画を行う処理方法である。

まず、データベースサーバーに保存されている視聴ログを取得し、Moodleに送るための常駐プログラムをMoodleのWebサーバー上で起動させておく。

次にMoodleのデータベースモジュールのテンプレートモジュールのヘッダまたはフッタに、必要な視聴ログが保存されているフィールドIDをMoodleのWebサーバーに送信するプログラムを作成する。これによりMoodleの当該データベースモジュールにアクセスした際、WebサーバーとMoodleの通信が開始され、指定したフィールドIDがWebサーバーに送信される。Webサーバーは、受信したフィールドIDを元に、常駐プログラムでデータベースサーバーから視聴ログを取得しMoodleに送信することができる。

しかしこの手法は、デーモンがMoodleの管理権限にないため、教師が自コース以外の視聴ログを受信できてしまうという問題点がある。フィールドIDはデータベースを作成した順番に沿って番号が割り振られているため他のデータベースのIDを推測することは容易である。またフィールドIDは、テンプレートモジュールを用い、教師自身が記述する必要がある。そのため教師がフィールドIDを書き換えることで、自コース以外の視聴ログを取得し、可視化グラフを作成することができてしまう。

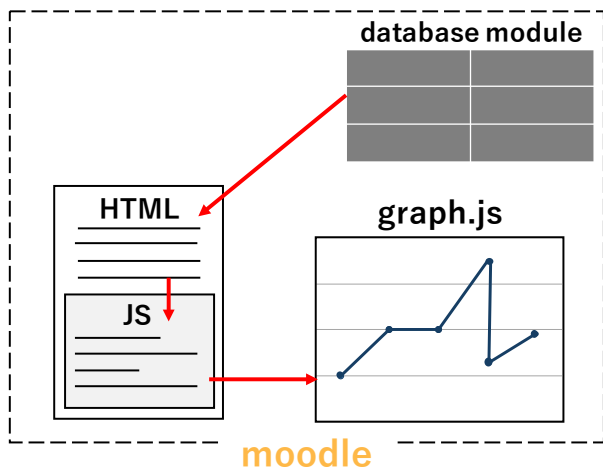


図 5 getElementByClassName 方式の概要

3.2. getElementsByClassName 方式

getElementsByClassName 方式は図 5 に示すように Moodle のデータベースモジュールにアクセスした際に HTML で表示されている視聴ログの一覧データから「getElementsByClassName 関数」を用いて視聴ログを取得し可視化グラフを作成する手法である。

データベースモジュールの初期設定の状態では、視聴ログを HTML から取得し可視化グラフとして利用するためのデータ構造に変換することが難しい。そのため、視聴ログをより取得・変換しやすい形にするために、一覧表示エン트리モジュールの繰り返しエントリを用い HTML でのデータ構造を変更する。例えば div タグを使用して、各エントリの class に UserID, id に EntryTime を設定する div タグの中には CurrentTime と ElapseTime を格納することで、どの学生のいつの視聴ログであるか区別することができ、可視化グラフに利用しやすいデータ構造として HTML 上に表示することができる。

またデータベースモジュールの初期設定では、最大で 1,000 件まで保存データを一覧表示することができる。学生が何も操作を行わず動画を見終えた場合、保存される視聴ログは、初めの「再生」時と視聴後の「ページ離脱」時の 2 つのみであり、500 回分の視聴データを一覧で表示することができる。しかし実際には多くの学生による様々な操作イベントによって 1000 件を超える視聴ログがデータベースに保存されることが想定される。そのため HTML から取得する視聴ログを元に可視化グラフを作成するこの手法では、1000 件を超えて保存された視聴ログも含めすべての視聴ログを HTML として表示する必要がある。そこで一覧表示するデータ数の設定が行われている「moodle/mod/data/lib.php」ファイルを用いて、想定されるデータ量に合わせた一覧表示数に設定を変更する

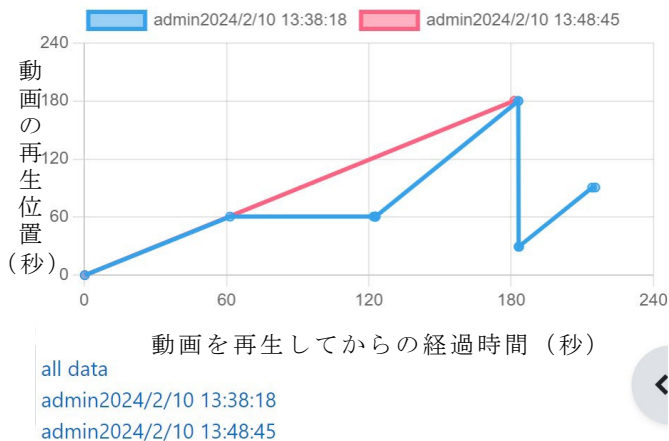


図 6 可視化グラフ

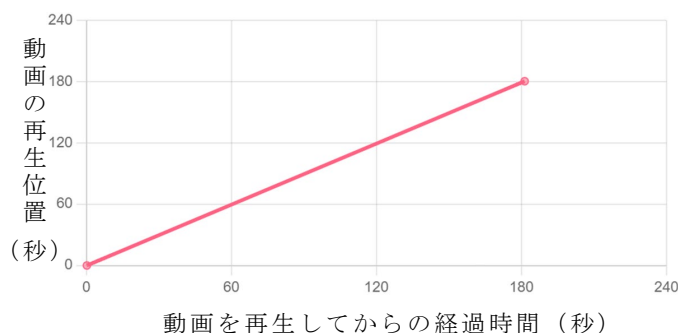


図 7 操作イベントが発生しなかった可視化グラフ

ことで 1000 件を超えるデータを一覧として表示することができる。

3.3. 比較

getElementsByClassName 方式の場合、デーモン方式の様に Moodle の管理権限下でないデーモンを用いる手法とは異なり、教師は、自コースのデータベースモジュールに HTML として表示された視聴ログのみを取得可能であるため、他のデータベースへの不正アクセスの問題はない。

また getElementByClassName 方式の場合、データベースモジュールへアクセスする際に、保存したすべての視聴ログを HTML で表示しなければいけない。そのため、アクセス時の処理が重くなり読み込み時間が長くなってしまふ。それに比べデーモン方式の場合、HTML で視聴ログを表示する必要が無く、バックグラウンドで必要なデータが取得されるためデータベースモジュールへの読み込み時間が getElementByClassName 方式と比べ短くなる。

4. グラフ化

データベースモジュールのテンプレートモジュールのヘッダまたはフッタに JavaScript ライブラリである「chart.js」とそのプラグインを用いて、図 6 に示す

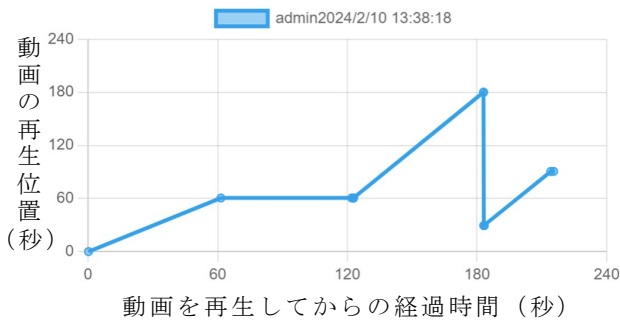


図 8 操作イベントが発生した可視化グラフ

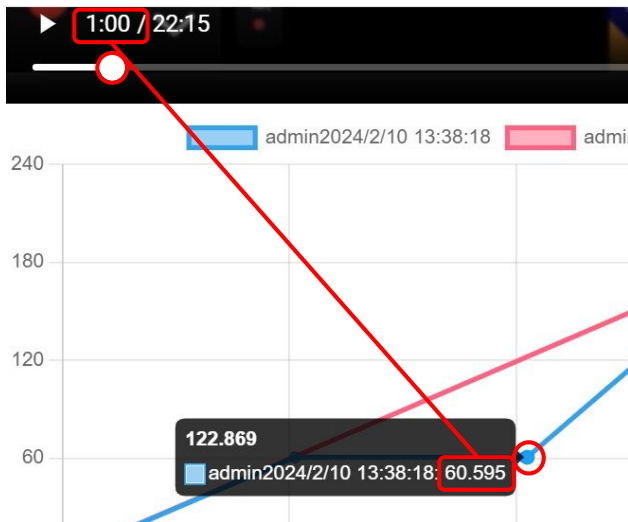


図 9 動画とグラフの再生時間の対応

ように縦軸を動画の再生位置，横軸を動画を再生してからの経過時間とした折れ線グラフを作成する。

折れ線グラフからは学生が授業動画に対して行った操作イベントを確認することができる。例えば図 7 のように学生が動画を再生してから動画が終了するまで全く操作を行わなかった場合は，動画の時間を示す縦軸と動画を再生してからの経過時間を示す横軸の値がほぼ同一の直線のグラフになる。図 8 のように学生が動画を一時停止した場合，動画の縦軸の変化はなく経過時間の値だけが大きくなる平らな折れ線グラフとなり，動画の飛ばし再生，または倍速を変更した場合は経過時間を示す横軸に対して，動画時間の値の変化が大きくなる。

また，データベースモジュールには動画を表示することも可能で，図 9 に示すように動画のシークバーと可視化グラフのマーカーを対応させることができる。これにより，可視化グラフのマーカーをクリックした際に，データベースモジュールに表示した動画の再生位置を可視化グラフの縦軸の動画の再生位置の値に変更させることができる。

さらに特定の視聴ログのみを可視化グラフとして表示したい場合図 6 の様なリンクを作成しオンクリッ

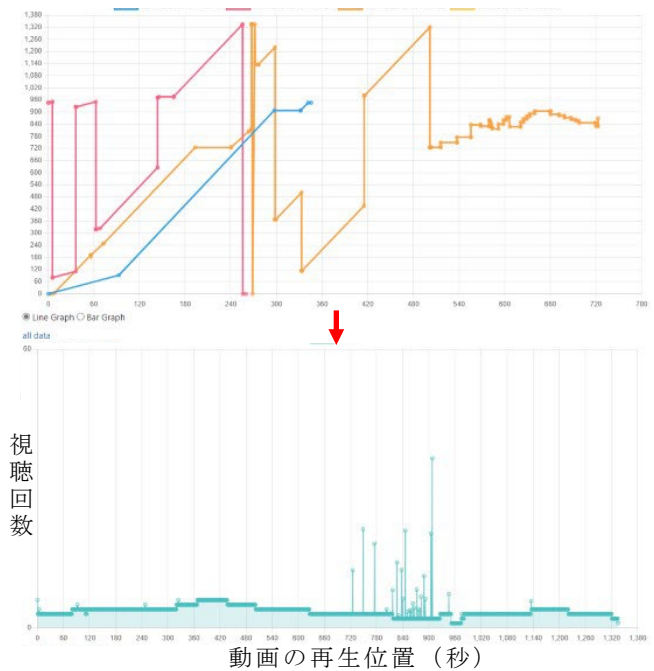


図 10 視聴回数のグラフ

クイベントを用い，取得するデータを選択することで，該当する学生のみの視聴ログを可視化グラフとして表示することができる。また図 10 に示すように動画を 1 秒間隔のセクションに区切った値を可視化グラフの横軸に設定し，動画の視聴回数を縦軸に設定することで，動画のどの部分の視聴回数が多いのか理解しやすいグラフを作成できる。

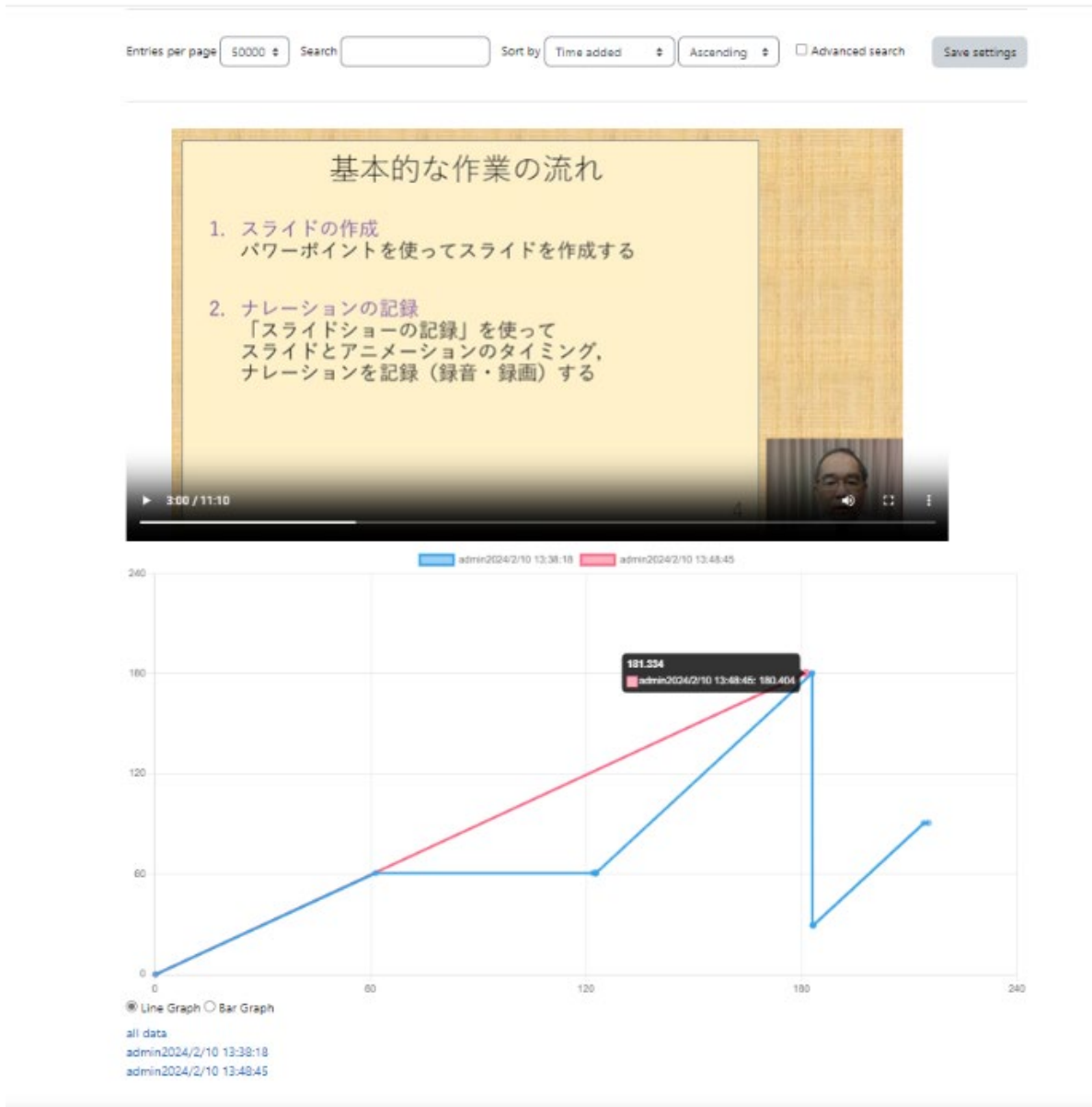
5. まとめ

本研究では，まず可視化グラフに必要な視聴ログの定義を行い，視聴ログを保存するタイミングについて 2 つの手法を検証した。Moodle を用いたオンデマンド授業では，多数の学生による動画視聴を考慮すべきであるため，可能な限りデータ数を抑えるため，視聴ログを保存するタイミングとして操作イベント方式が適切であると考えられる。また，保存した視聴ログの取得方法について，操作イベント方式と合わせて `getElementsByName` 方式を利用することで，少ないデータ数でさらに，デーモン方式と比べ比較的に安全に可視化グラフを作成することができる。

文 献

- [1] 清水健一，藤本茂雄，松本暢平，檜垣泰彦：Moodle 用動画配信システムの内製とコロナ禍における運用・評価，電子情報通信学会論文誌(D)，J105-D (10)，pp572-583(2022 年 7 月)
DOI: 10.14923/transinfj.2021LIP0015
- [2] 岡田浩希，鈴木 孝幸，納富 一宏：映像講義における受講者の視聴ログの可視化，第 29 回バイオメディカル・ファジィ・システム学会年次大会 講演論文，29 (0)，pp.182-185(2016 年 11 月)
- [3] 斎藤準：Moodle 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，pp11-26

付録



実際に開発した Moodle のデータベースモジュールにおける可視化グラフのスクリーンショット