

反転授業を継続的に行うためのスクリプトによる 動画制作システムの開発

村上智哉[†] 檀浦紗弥[†] 井上繕[†] 沖津真歩[†] 下見真生[†] 長尾和彦[†]
弓削商船高等専門学校[‡]

1. はじめに

高等教育において学生が主体的に学ぶことを促進する教育方法として、反転授業がある。これは、学習者が新たな学習内容を自宅などで動画により事前学習することで、教室での授業を知識の確認や問題解決学習、応用的な問題を解く時間などにあてることができる^[1]。

最近では AviUtil や iMovie などの動画作成ソフトや AdFlow などの制作管理システムがあり、高品質な動画の制作環境を個人でも用意することができる。また、動画教材の配信環境として Moodle などの E-learning システム、Youtube など動画配信システムの発展により、教員から学生へ動画提供のハードルは下がりつつある。

しかし、反転授業の問題点として①動画制作スキルや環境、②自身の授業目的に沿った動画を用意する時間、③受講学生に合わせて動画を適宜修正する時間が教員側に必要なことから、教育現場での導入は進んでいない。そこで、本研究では教員の負担を少なくし継続的に反転授業を行うための仕組みを検討する。

2. ヒアリング

本研究を進めるにあたり、本校で反転授業を導入している教員へ「現状の動画制作環境への不満点」についてヒアリングをした。結果を以下に示す。

2.1. 動画制作ソフトが難しい

動画のワイプ、音声差し替えなどの処理をマウス操作やショートカット入力などをする必要がある、ソフトが提供する高度かつ複雑な機能は必要なく、よりシンプルなもの望ましい。

2.2. テンプレートが作れない

動画教材ではリアルタイムフィードバックができないため、単元の目標に合わせて 1 授業ごとの教材を目的と内容を十分に設計する必要がある。

る^[2]。このことから統一感ある構成で 1 授業ごとの動画を作りたい。

2.3. バージョン管理ができない

動画は複数の動画や画像ファイルを利用する関係上、編集時のファイルが大きくなりがちのため編集記録を残しづらい。

3. 動画編集システムの実装について

前述のヒアリングから、テンプレート作成とバージョン管理を主軸とした動画制作システムが必要なが確認された。

3.1. システム概要

動画素材の組み合わせをテキストで記述し動画生成を行う。またサーバ上でバージョン管理システムを導入することで、持続的な動画編集を可能とする。

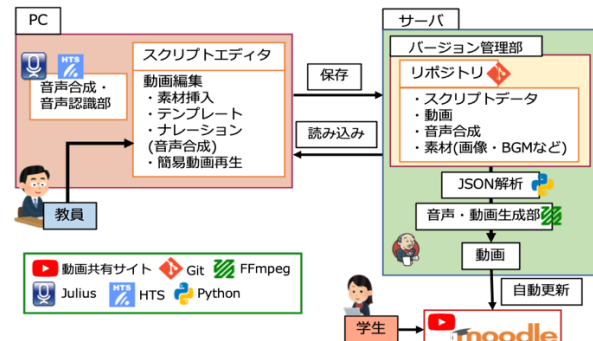


図 1. システム構成図

3.2. 動画編集機能

テキストエディタを用いてスクリプトと呼ばれるテキストを記述することで、動画編集を行う。動画の生成は、スクリプトを Lark^[3]でパースし、その結果を FFmpeg^[4]で処理することにより実現する。表 1 にスクリプトの定義を示す。

3.3. ナレーション機能

合成音声によるナレーションを動画に挿入することができる。音声の合成には OpenJTalk^[5]を使

A Development of Video Production System based on Script for Continuous Flipped Classroom

[†] Tomoya Murakami, Saya Danura, Zen Inoue, Maho Okitsu, Maiku Shimomi, Kazuhiko Nagao

[‡] National Institute of Technology, Yuge College

用する.

3.4. テンプレート機能

学生にわかりやすい授業を提供するため, PREP 法に基づいたテンプレートを提供する. PREP 法は要点・理由・具体例・要点の4つの要素の順番で文章などを構成する構成法である.

3.5. 管理・更新機能

動画素材ファイルと生成したスクリプトをサーバ上の Git リポジトリで管理することで, バージョン管理が可能である. また, リポジトリを公開することで, 共有が可能である.

4. システムの評価

本システムの開発後に本校の教員・学生・企業関係者からシステムの評価を受けた. 本校の教員・学生からは, 自分で喋らないため音声を録り直す必要がないのは嬉しい, わずか数分で動画ができた. 企業関係者からは, 社員育成の動画を作成する時に使いたい, 簡単にできるため教材以外にも使えそう, など意見があった.

5. 今後の課題

本システムを実用化する際, いくつかの課題が確認されている. 編集した動画を確認する際に時間がかかる点である. 原因として, 動画の生成に時間がかかっていることが挙げられる. これは動画再生のために動画の全部分を生成しているために発生している.

この問題を解決するために, スクリプトの部分コンパイル機能が必要であると考えられる. この機能を実装することで, 確認する部分のみを生成できるため動画の確認にかかる時間を削減できると予想する. スクリプトによる編集のため, 機能の実現は容易である.

6. おわりに

本研究では, 教員が動画制作をする際の制作負担を軽減するようなシステムを開発した. 今後は実際の授業で使用してもらい, 様々な科目のデータを参考にシステム改良に努めたい.

参考文献

- [1] 文部科学省: Blended Learning, pp69, 2013.8.26
- [2] 鈴木克明: e-Learning 実践のためのインストラクショナル・デザイン, 日本教育工学会論文誌 29(3), 2006.2.20
- [3] Main Page - Lark, <https://lark-parser.readthedocs.io/en/latest/>, 2020.1.8
- [4] FFmpeg, <https://www.ffmpeg.org>, 2020.1.8
- [5] Open JTalk, <http://open-jtalk.sourceforge.net>, 2020.1.8

表 1. スクリプトの定義

```
?suite: (_NL|stmt)*

stmt: NAME "=" small_stmt _NL -> assign_stmt
      | small_stmt _NL -> seq_content
      | "&" small_stmt _NL -> para_content
      | "{" (_NL|stmt)* "}" -> block
      | "!" CMDLINE -> cmd

small_stmt:
  | "Video(" FILENAME ")" [ "." parameter ]* -> video
  | "Sound(" FILENAME ")" [ "." parameter ]* -> sound
  | "Image(" FILENAME ")" [ "." parameter ]* -> image
  | "Text(" FILENAME ")" [ "." parameter ]* -> text
  | FILENAME [ "." parameter ]* -> text
  | NAME [ "." parameter ]* -> var

parameter:
  | "start(" NUMBER ")" -> start
  | "end(" NUMBER ")" -> end
  | "position(" NUMBER NUMBER ")" -> position
  | "play(" NUMBER ")" -> play
  | "soundoff" -> soundoff
  | "soundon" -> soundon
  | "fadeout" -> fadeout
  | "fadein" -> fadein
  | "title" -> title
  | "telop" -> telop
  | "voice" -> voice

%import common.CNAME -> NAME
%import common.INT -> NUMBER
%import common.WS
%import common.ESCAPED_STRING -> FILENAME
%import common.WS_INLINE

%ignore WS
%ignore WS_INLINE
%ignore COMMENT

COMMENT: /#[^\n]*/
_NL: ( \r?\n[\t ]* / | COMMENT )+
CMDLINE: /#[^\n]*/_NL
```