

遠隔授業における YouTube の利用

YouTube in Remote/Online Classes

八尋剛規*

Takeki YAHIRO

1. はじめに

2020年、コロナ禍の影響で遠隔授業が急遽実施された。動画配信の手段として複数の選択肢があったが、筆者の場合はYouTube[®]あるいはYouTube Liveを利用した。これは、次の理由による。1) 学習者(学生)は、動画配信サービスとしてYouTubeに慣れており、利用開始にあたり学習者の不安感を軽減できる、2) Webサイトに動画を埋め込むことが容易であり、さらにJavaScriptから利用できるYouTube IFrame API¹⁾(以下、API)により細かな制御ができる(他の動画サービスもJavaScript APIは公開されているが、絶対的な利用数が少なく情報が少ない)、3) 音質、画質が他のサービスに比較し良好である、4) ライブ配信中でも動画の一時停止、タイムシフト再生などが可能で、学習者の様々な学習スタイルに柔軟に対応できる、などである。

さらに動画配信だけでなく、授業進行に合わせ関連するHTMLコンテンツ(以下、「e 黒板」と称する)を、動画とは別に表示させることとした。これは次の理由による。1) ネットワーク環境によっては十分な動画の画質が確保できず動画中の文字などの視認性が低下する恐れがある、2) 授業中に視聴させたい動画がある場合、その動画を配信動画中に含めると著作権上の問題が生じるが、HTMLとして受講者画面に埋め込むことで解決する、3) HTMLだけでなく、JavaScriptの利用により、本来のWebブラウザの機能を限りなく活かすことができ、双方向授業の実装やPowerPointなどのプレゼンテーションソフト以上の学習効果が期待できる、4) 受講者は授業ページのみで授業を完結することができる(ページ遷移を伴うと学習者がネット上で迷子になってしまう可能性がある)などである。

これを実現するには、動画の再生タイミングとe 黒板表示のタイミングを合わせる必要がある。筆者は従来から授業においてWeb上に構築しているコンテンツを用いてプレゼンテーションなどを行っている。また、そのための授業用Contents Management System(CMS)²⁾もオリジナルで開発し利用している。2020年の遠隔授業に対応するため、本CMSに遠隔授業を実現するためのいくつかの機能を付け加えた。本稿では、その機能の一部である、動画表示とe 黒板のタイミング調整の手法につ

いて述べる。

2. 実装

ライブ配信とオンデマンド配信とで実装方法が異なるため、個別に述べる。

2.1 ライブ配信 授業者は、ライブ配信を行いながら、予め用意したHTMLコンテンツをクリックすることで特定領域のコンテンツがサーバに送られる(付録A)。学習者の端末は一定間隔でサーバにアクセスし、e 黒板コンテンツを取得し表示する。このとき、動画が実際に学習者の画面に表示されるタイミングとe 黒板コンテンツが表示されるタイミングが異なる。動画は最低でも5秒の遅延が生じるため、この遅延を考慮したタイミング調整が必要である。ライブ配信(YouTube Live)の場合、送信側で遅延設定を三段階から選択することができるが、今回「超低遅延」(実質値5秒程度)を選択した。しかし、この遅延は授業の経過時間とともに変化し、また学習者の端末ごとに異なることもあり、通り一遍の対応は不可能であった。そこで、学習者側の画面にタイミング調整のボタンを設け、学習者側に調整を任せることとした。ログの解析の結果、全体の6%でタイミング調整が行われていた。

なお、ライブ配信の場合も、ライブ配信を録画した動画を後日視聴できるようにしているが、その際のe 黒板表示は次に述べるオンデマンド配信方式を利用している。



図1 配信画面の例(学習者画面)

上段がYouTubeを利用した動画、下段はHTMLコンテンツ(e 黒板)

*経営学部観光ビジネス学科教授

2.2 オンデマンド配信 オンデマンド配信の場合、学習者側端末で再生中の動画に同期し、対応する e 黒板コンテンツをサーバより得て、表示する機能を設けた。同期して表示するには、動画再生時の経過時間が必要となる。事前調査の結果、一部の学習者は動画を一時停止や早送り、1.2 倍速あるいは 1.5 倍速で視聴していることが判明した。よって学習者端末の経過時間と動画の経過時間が同一とは限らない。API によりこれらを制限することもできるが、オンデマンドの利点が失われることにつながりかねない。そこで、再生中の動画の経過時間を、API を利用し一定間隔で取得することとした。これにより、学習者が動画の再生速度を変更した場合や、一時停止、巻き戻し、早送りなどの操作に対しても、e 黒板の表示タイミングを合わせることが可能となる。

一方、サーバ側には、次のようなデータベースがあり、動画ごとの e 黒板コンテンツの表示タイミングと内容を管理している。

表 1 e 黒板管理テーブル

列名	型など	用途
youtubeid	varchar(20)	YouTube 上のビデオを識別するための ID
elapsed_time	int	経過時間 [秒]
html	text	HTML コンテンツ

※関係ある列のみ掲載

学習者が受講を始める際、その受講に必要な e 黒板の表示タイミングの一覧 (表 1 における elapsed_time の行列) を取得する。この時、HTML コンテンツも事前に取得する方法もあるが、学習者端末のメモリーを圧迫する懸念があったため、随時取得する方法とした。

学習者端末で受講が始まると、1 秒間隔で動画の経過時間 (秒数) を API の getCurrentTime() メソッドで取得する。この値と先に取得したタイミング行列と比較し、必要な e 黒板コンテンツをサーバより Ajax で取得し表示する (付録 B)。

また、次の e 黒板コンテンツに切り替わるまでの時間を、動画と e 黒板の間にプログレスバーで表示し、e 黒板コンテンツ切り替えのタイミングが学習者に視覚的にわかるようにした。これは、e 黒板が突然前触れもなく内容が変更されることにより、学習者の操作 (ノート作成やコピー&ペースト) の妨げにならないようにするためである。

3. まとめ

2020 年 4 月以降の授業実践の結果、学習者からは「一つのページ内で学習が完結するのでわかりやすい」、e 黒板領域に投票機能などを表示することにより簡易的な双

方向の授業も行えることから「みんなで授業に参加している感じがする (孤独感が薄れる)」という評価や感想も得られている。今後も学習者からの意見を随時とりあげ改良を行なっていく予定である。

参考文献

- 1) YouTube IFrame Player API, (URL)
https://developers.google.com/youtube/iframe_api_reference
- 2) 八尋剛規：授業運営に要求される機能検証用 CMS ベースシステムの開発と授業実践, 日本産業技術教育学会九州支部論文集, Vol. 27 pp. 1-9,
(URL) http://www.jste.jp/kyushu/ronbun_paper/jste-kyushu-journal-27.pdf

付録 A

```
var $e = $(this); // クリックした要素
var html = $e.prop('outerHTML');
$.nextUntil('h1,h2,h3,h4').each(function(){
    html += $e.prop('outerHTML');
});
// その後、html の内容を Ajax でサーバに送信
// サーバ側で表 1 のテーブルに保存される
```

付録 B

```
function checkQue()
{
    var i, t, n; // que::array はコンテンツのタイミング行列
    if( youtube_target != undefined ){ // YouTube Object
        // 動画の経過時間の取得 (秒) YouTube IFrame API
        t = parseInt( youtube_target.getCurrentTime() );
        n = que.length;
        for( i = 0 ; i < n-1; i++ ){
            // 経過時間 t における e 黒板のコンテンツを検索
            if( que[i] <= t && t <= que[i+1] ){
                if( checkQue.current != que[i] ){
                    checkQue.current = que[i];
                    checkQue.timespan = que[i+1] - que[i];
                    $('#debug-que').html( checkQue.current );
                    $('#debug-span').html( checkQue.timespan );
                    // ajax でサーバより取得
                    $.ajax( {
                        // 対応するデータのリクエスト
                    }).done( function( txt ){ // 表 1 の 1 レコード JSON
                        let r = JSON.parse( txt );
                        $('#ebb').html( r.html ); // e 黒板領域に出力
                    }); // エラー処理は省略
                }
                break;
            }
        }
        if( checkQue.timespan > 0 ){
            // e 黒板の表示が切り替わるまでの残り時間を表示
            let span_elapsed = t - checkQue.current;
            let remain = checkQue.timespan - span_elapsed;
            let width = Math.round( remain/checkQue.timespan * 100 );
            $('#ebb-show-remaintime').css( { width: width + '%' });
        }
        timerid = setTimeout( function(){
            checkQue();
        }, 1000 );
    }
}
```

※一部の処理及び変数の宣言とエラー処理は誌面の都合により省略