

iPadを用いた授業づくり —Webアプリ作成を中心として—

長田 朋之（文教大学教育研究所客員研究員）

今田 晃一（文教大学教育学部）

Structure of Classes Using Tablet Terminals (iPad) -Focus on Preparation of Web Applications without Depending on OS-

OSADA TOMOYUKI, IMADA KOICHI

(Guest Researcher Institute of Education, Bunkyo University)
(Faculty of Education, Bunkyo University)

要旨

文部科学省「教育の情報化ビジョン」に示された、特定の端末やOSに限定することなく利用できるタブレット端末用アプリケーションの簡易な手法を確立するため、HTML、JavaScript、Perlを用いたWebアプリを試験的に開発した。またそれを利用した小学校での授業実践を通して検証を行い、今後のOSに依存しないアプリ開発の課題を明らかにした。

1. はじめに

平成23年4月28日に文部科学省は、2020年度に向けた教育の情報化に関する総合的な推進方策と位置付けた「教育の情報化ビジョン」を公表した。その中の「21世紀にふさわしい学びの環境とそれに基づく学びの姿(例)」では、具体的に児童・生徒が携帯端末やデジタルノートといったタブレット型電子情報端末（以下、「タブレット端末」と表記する）を利用しながら学習活動を行う場が示されている¹⁾。また、文部科学省の「学びのイノベーション事業」などを通じて、実際の教育現場にタブレット端末が導入される事例も増えてきており、今後、タブレット端末の導入が学校現場において着実に進むことが予想される。

佐賀県では、2011年度に「先進的ICT利活用教育推進事業」を開始し、2013年度4月に36校の県立高校全てで情報端末の活用を始める計画だったが、導入時期が1年間延期されて2014年度に変更となった。この1年間

の延期はタブレット端末の機種選定のためであることを、佐賀県教育庁・教育情報化推進室は、2013年5月16日、教育機関向けの展示会「教育ITソリューションEXPO」の専門セミナーで明らかにした²⁾。

以上のようにタブレット端末は、その導入には教育現場も前向きであるが、どの機種を選ぶかにはどのOS（Operating System:基本ソフト）に依存するかを限定することであり、タブレット端末の機種選択において重要な問題となっている³⁾。

そこで本稿では、まずタブレット端末のOSについての現状を述べ、次にOSに依存しないアプリを開発した実践について報告する。

2. タブレット端末とOS

現在タブレット端末は世界中のメーカーから発売されているが、ハードウェアは汎用的な部品が使われているので、画面の大きさ・タッチパネル・重量といった基本的な性能に大きな差は見られない。一方、タブレット端

末に搭載される基本的なソフトウェアであるOSは種類が限られており、Appleが開発したiOS、Googleが開発したAndroidの2つが主要なOSであるという状況である。2012年度通期に日本国内で出荷されたタブレット端末における各OSのシェアは、株式会社MM 総研の調査において、iOSが52.5%、Androidが42.4%、その他が5.1%となっている。

現時点では、iPad等が搭載しているiOSがトップシェアであるが、数年後も同じである保障はなく、場合によっては市場競争に敗北して消滅したり、企業の方針によって開発が停止したりする可能性もある。仮にiOSの入手ができなくなり、それまでに導入していたiOS用のアプリのデータをAndroidやWindowsなどのOSを搭載した端末にコピーしても、プログラムの形式が異なるためにアプリを起動することはできない。つまり現状では、特定のタブレット端末を導入してアプリを購入した場合、数年後の端末更新時に、そのアプリを継続的に使用するためには同系統のOSを搭載した端末を再び購入する以外に手段はなく、異なるOSに変更した場合はそれまでに購入したアプリをすべて放棄した上で、新しいアプリを購入することになる。

これまでの紙を基本とした教材では、教師が自作の教材を作成し、授業で配付することもあった。タブレット端末においても導入が進めば、タブレット端末向けの教材を教師が自作するようになる可能性がある。そのとき、特定のOSに限定されたタブレット端末の利用環境になっていると、現場の実態にあわせて教師が作った教材が勤務校の変更やタブレット端末の更新によって利用できなくなる可能性もある。またそれまでに作成した教材を使い続けるためにより優れたOSに切り替えることを躊躇する、という状況が想定される。タブレット端末導入に関して、前述の佐賀県のように機種選定に対して慎重になる理由は、各OSにおけるアプリケーション（以下、「ア

プリ」と表記する）の互換性がなく、端末にインストールできるOSも基本的に自由に選ぶことができないので、端末を選定するとOSと学習用アプリの組み合わせも半ば強制的に決まってしまうためと考えられる。

また、タブレット端末を使用する場面は、校外に持ち運んで使用したり、グループ活動で1台の端末の画面を全員で閲覧したりといった様々な場面が「教育の情報化ビジョン」の中でも想定されている。それらのすべての場面を単一機種の端末で運用することは難しいと予想できる。将来的には複数の端末を場面に応じて切り替える運用となる可能性がある。またさらに将来的には、台湾で一部実験的に取り組まれている端末フリーの学習環境も視野に入れる必要がある。

そのような場合に、複数の端末で同一の学習用アプリが利用できないと、操作法の異なる類似のアプリを使い分けるなど、システムが複雑になって円滑な運用が困難になる。そのため、動作環境が特定の端末やOSに限定されず、長期的に利用し続けられるアプリの開発手法を今から確立していくことが、緊要性のある課題として明らかになってきている。

3. iPod touchの試験的導入

タブレット端末を学校に本格的に導入し、アプリ開発を行っていく前段階として、タブレット端末を教育活動に取り入れた際に起こる効果と課題を前もって把握するため、少数のタブレット端末を導入して試験的な運用を行った。

タブレット端末としてAppleの第4世代iPod touchを12台導入し、小学校3～6年生を対象とした授業等の教育活動の中で試験的に使用した。iPod touchはiPadよりも小型であるが、Wi-Fi、カメラおよびタッチパネルといったタブレット端末としての主要な機能はiPadと同じであり、OSも同一iOSのバージョンのものを利用することができる。

(1) 理科の実験観察で使用

3年生の温度計で温度を測定する実験で、一般的な棒温度計を目視する方法による温度測定とともに、温度計の表示を客観的な方法で記録して、温度測定が正しく行われたか検証するために3～4人の班ごとに1台ずつ合計12台のiPod touchを配付して温度計の目盛りを撮影させた(図1)。

どの班も温度計を目視で測定した後、iPod touchのカメラで目視した場合と同じように撮影できる向きを探して調整を何度も行い、目視と撮影を時間をかけて観察した。従来の目視だけで児童に観察させる方法では、どの班も1回目盛りを読んだだけで「正しく読めた」と判断し、温度計の垂直方向から目視できたことを確認するとか、目盛りを読み間違えていないか確認するとか、測定方法の検証を十分に行わない児童の多さが課題であった。しかし今回は撮影内容の確認を繰り返す中で自然に目視を繰り返したり、班員同士で測定方法の検証を行ったりすることができていた。

また、班員とあまり協調せず実験操作を一人で進めてしまう児童も、一人では実験と撮



図1 iPod touchを活用した3年生の温度計の撮影

影の両方ではできないので班員と協力するようになった。実験が不得手なために実験に消極的だった児童も、撮影する役が与えられ、実験に参加するようになった。撮影後は班員全員が興味を持って撮影内容を確認し、内容について互いに意見を出し合う姿が見られた。

一方、次の点が課題として明らかになった。それは、児童が端末の設定を変更できてしまう、撮影データの取り込みに時間がかかる、撮影データの消去に時間がかかる、不要なアプリを起動できてしまう、の4点であった。この中で「児童が端末の設定を変更できること」と「不要なアプリを起動できてしまう」についてはiOSの標準機能である「学習サポート」によって制限をかけることができた。

「撮影データの取り込みに時間がかかる」についてはiOSの標準機能である「iCloud: フォトストリーム」によって自動的にパソコンへ取り込むことはできるものの、撮影と同時に転送されるわけではなく、転送されるタイミングがわからないので、授業中に児童が撮影した内容を教師が把握する用途では役に立たなかった。また、USBケーブルでパソコンと接続することで外部ストレージとして認識させることができるので、USBハブを利用して12台の端末を同時に1台のパソコンに接続することを試みたが、複数の端末を正常に認識できなかったため、1台ずつUSBでパソコンと接続してデータをコピーしなければならなかった。

「撮影データの消去に時間がかかる」という課題については、同一学年の複数の学級に対して同じ内容の授業を展開するために前の授業で撮影したデータは削除する必要があったが、端末の写真を削除するために1台あたり1分程度の時間をかけて手作業で行う必要があり、休み時間の多くを写真の削除に費やした。授業時間中に児童に写真を削除させることも可能だが、学習時間をそのような用途で使用することは本時の目標との関連からも

望ましくないので、教師が端末上の撮影データを一括消去できる機能の必要性を感じた。

(2) フィールドワークにおける利用

校外学習で見つけたものを記録し、地図上にまとめる学習活動の写真撮影のために7～8人の班に1台ずつiPod touchを配付して撮影させた。落下防止のために首から下げる方式の市販のストラップを装着した(図2)。



図2 iPod touchを利用したフィールドワークにおける撮影

午前8時から午後3時まで約5時間に渡って使用したが、12台とも電池切れになったり保存領域不足になったりすることなく最後まで正常に作動した。iPod touchの破損を懸念していたがストラップが外れて落下したり、水没したりといった事故は発生しなかった。

一方、課題としては、撮影データの取り込みに時間がかかる、撮影内容をリアルタイムに把握できない、撮影データの消去に時間がかかる、という3つの課題については前述の通りである。「撮影内容をリアルタイムに把握できない」については、「iCloud：フォトストリーム」を用いて児童の撮影内容を把握することを想定して、校外に小型のバッテリー内蔵可搬型アクセスポイント(モバイルWi-Fiルーター)でWi-Fiを構築することを検討した。しかし最大同時接続台数13台以上を保証しているモバイルWi-Fiルーターを入手できなかったため、児童用iPod touch

12台と教師用iPad mini 1台を接続できる環境は残念ながら構築できなかった。

4. アプリ開発言語と開発環境

本研究の目的は、特定のOSに限定されることなく利用できるタブレット端末用アプリの開発手法を確立することである。そのためには、どのOS上でも実行可能なプログラミング言語を使用する必要がある。そこで、今回はタブレット端末上で共通して実行できる言語として「HTML」と「JavaScript」を選択した。この2つの言語を選択した理由は、OSに関係なく利用できる、仕様が公開されている、利用によるライセンス料が発生しない、広く一般で利用されている、15年以上の長い歴史を持っている、以上の5つである。

HTMLとJavaScriptはどちらもインターネット上のWebページを構成する言語である。HTMLは1993年にインターネットで利用される技術の標準化を策定する組織である「IETF」によって最初の仕様が公開された20年以上の歴史を持つ言語である。

一方、JavaScriptは情報通信システムの分野における国際的な標準化団体である「Ecma International」によって1997年に仕様が確定された言語で、HTMLには及ばないものの16年以上の歴史を有している。どちらの言語も仕様が世界に向けて一般公開されており、無償で自由に利用することができる。iOSとAndroidのどちらでもHTMLとJavaScriptで作られたアプリ(以下、「Webアプリ」と表記する)を実行することができる。HTMLとJavaScriptはインターネットの中核技術であるため、今後発売される端末・OSにおいてもインターネットに接続できる端末であればこの2つの言語に対応するものと考えた。

Webアプリはタブレット端末にインストールせず、サーバ上にアプリを設置して各タブレット端末のブラウザから必要に応じて

Webページとして呼び出して実行するように設計した（図3）。

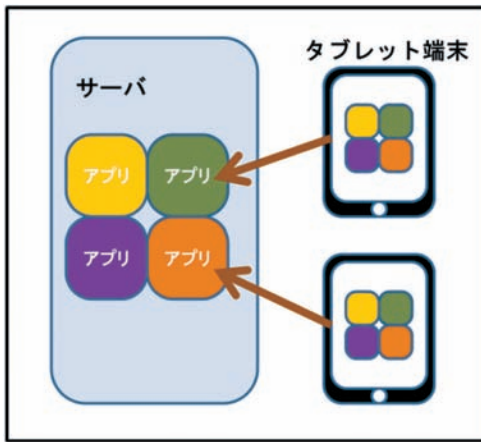


図3 サーバ上のアプリの呼び出し

一方、Webアプリは、一般的には端末内の演算装置が直接に演算処理を実行するネイティブアプリと比較した場合に不利となる点も多い。その代表的なデメリットとしては、アプリの実行速度が劣る。端末を制御するアプリは作れない。Webアプリは比較的に高速に実行されるが、ネイティブアプリはCPUで実行しやすい形式に事前にコンパイルをしているので、実行速度でWebアプリは劣る。Webアプリでは端末を制御することもできない、という点である。例えば、画面の明るさを変更したり、内蔵時計を変更したりといったハードウェアの設定を変更するようなアプリは作れない。ただし、カメラ・GPS・加速度センサーなどの一部にHTML5とJavaScriptを用いて接続し、情報を読み出すことはできる。

Webアプリはネイティブアプリと比べて制限はあるが、端末固有の機能を利用することは特定の端末にアプリの実行環境を限定することになるため、本研究の主旨として端末固有の機能は利用しないので、Webアプリの持つOS互換性の方が有効であると判断した。

iPod touchの試験的導入で明らかとなった課題である「撮影データの取り込みに時間がかかる」と「撮影データの消去に時間がかかる」に対応するため、今回のアプリ開発ではタブレット端末上にデータを保存せず、すべてサーバ上にデータを保存する設計とすることで対応した。サーバ上にすべてのデータを保存することで、データを端末からコピーしたり、消去したりといった作業は不要にできる。

サーバ上でデータを保存したり読み出したりといった処理を行うプログラムは、Perl言語を用いた。Perl言語は25年以上の歴史を持つオープンソースのプログラミング言語である。C言語などと同様に広く利用され、サーバ用として用いられるUNIX・Linux・Windowsなどの主要なOS上で実行することができる。Perl言語はインタプリタ型言語であるため、アプリを起動するたびにソースコードを解析して記述されたプログラムに沿って逐次実行される。C言語のようなコンパイル型言語はソースコードを事前にCPUが直接実行できるマシン語にコンパイルして、速やかに実行できる状態に変換された状態でプログラムが起動されるので、一般的にインタプリタ型言語はコンパイル型言語と比べてプログラムの実行速度や負荷の面で不利である。

しかし、コンパイル型の言語はコンパイル後に異なるOS上で実行したり、CPUのアーキテクチャが変更になったりした場合、そのサーバ環境用に再度ソースコードからコンパイルし直す必要がある。インタプリタ型言語は、その言語に対応した環境であれば、OSやCPUのアーキテクチャの影響を受けることなく同じプログラムをそのまま利用することができる。今回は、特定のOSに限定されない普遍的なアプリ開発を目的としているため、同一のプログラムを変更せずに多くの環境で実行することができるインタプリタ型のPerl言語を選択した。

5. アプリの開発

(1) アプリ配信用プログラム

このプログラムは、サーバ上のアプリを管理して、必要に応じてタブレット端末へアプリを配信するためのものである。主な機能は「アプリの一覧表示」と「アプリの実行制限」である。

サーバ上にインストールされたアプリはこのプログラムによって自動的に認識され、管理者権限を設定された教師用のタブレット端末上にアプリの一覧として表示される (図4)。



図4 アプリ起動制限画面

教師は起動制限機能によって、児童用タブレット端末で起動できるアプリを直感的に制限することができる。また、このアプリを介在させることで、タブレット端末に新たなアプリを追加する場合もサーバ上にアプリをインストールするだけで各タブレット端末から利用できるようになる。

アプリ起動の制限と解除は次のプログラム (Perl言語) によって行った (図5)。

```

if ($in['app_mode'] eq "on") {
    unlink "../$in['app_name']/lock.txt";
} else {
    if ($in['app_name'] ne "設定") {
        open(DATAFILE, ">",
            "../$in['app_name']/lock.txt");
        print DATAFILE "";
        close(DATAFILE);
    }
}

```

図5 アプリ移動および制限と解除のプログラム

起動制限をかける場合は、そのアプリが保存されているディレクトリ (フォルダ) 内に「lock.txt」という名前のファイルを作成する。起動制限を解除する場合は「lock.txt」を削除する。

アプリが保存されているディレクトリ内に「lock.txt」という名前のファイルが存在するか確認を行い、存在する場合は起動制限がかけている旨を表示してプログラムを終了する。このプログラムを配信用アプリではなく、個々のアプリに埋め込む理由は、配信済みのアプリに対して起動制限をかけるためである。

(2) 写真・ビデオ撮影アプリ

このアプリは各タブレット端末で画像または映像を撮影して保存するためのものである。タブレット端末で撮影した画像・映像をサーバ上に保存することが主な機能である。

撮影ボタンをタッチするとカメラの撮影画面が起動し、撮影完了後、「保存した写真・ビデオの一覧」に追加される。撮影ボタンはHTML言語で図6のようにして表示させている。

```

<form>
  <p class="file">
    <input type="file" name="upfile"

    accept="image/*;capture=camera"
      id="file" />
  </p>
  <p id="dummy"></p>
</form>

```

図6 撮影した画像・映像をサーバ上に保存するためのプログラム

「input」タグの属性「file」によってファイルをサーバ上へアップロードする機能を有効化している。「form」タグにアップロード先サーバの属性指定がされていない理由は、ファイルのアップロード処理をJavaScriptの非同期通信によって行うためである。非同期通信アップロードは2つのJavaScriptライブラリ「jQuery.js」と「jQuery.upload.js」によって実現している。

タブレット端末からサーバへ送信する写真・ビデオのデータはHyper Text Transfer Protocolで転送される。サーバにデータが到達すると、ファイル受信用のCGIが起動して受信を行う。CGIにはPerl言語のcgi-lib.plライブラリを用いた。

保存した写真・ビデオの一覧を表示する際に、元の画像データをそのままサーバから端末へ転送すると、大量のデータがネットワーク上に流れ、ネットワーク全体に大きな負荷がかかるので、撮影された画像をサーバ上に保存すると同時に、圧縮した画像をキャッシュとしてサーバ上に保存し、端末へ転送することで負荷の低減を図った。図7がそのためのプログラム（Perl言語）である。

```

use GD;
use MIME::Base64;
$thumb_size = 140;
GD::Image->>trueColor(0);
$src = GD::Image->new($src_file);
      (省略)
$thumb = GD::Image->new($thumb_width,
  $thumb_height, 1);
$thumb->copyResampled($src, 0, 0, 0, 0,
  $thumb_width, $thumb_height, $width,
  $height);
$binary = $thumb->jpeg(60);
$base64 = encode_base64($binary, '');

```

図7 圧縮した画像をサーバ上に保存し、端末へ転送するプログラム

Perl用の画像編集ライブラリ「GD」とバイナリデータを英数字コードに変換するライブラリ「MIME::Base64」を使用した。Perl用のライブラリとしてはどちらも一般的なものである。

GDライブラリを用いて、幅140ピクセル、圧縮率60%のJPEG形式に圧縮する。このバイナリデータはMIME::Base64ライブラリで文字列に変換され、キャッシュとしてサーバに保持される。

圧縮してキャッシュとして保存されている画像データを呼び出す際は、以下のプログラム（Perl言語）が実行される（図8）。

```

foreach $cache (@caches) {
  $buffer = substr($cache, 0, $len);
  if ($buffer eq "$file_stat") {
    ($file, $file_size, $file_time,
      $base64) =
split(/</>/, $cache);
    $img_data = "data:image/jpeg;
      base64, $base64";
    $make_cache = "";
    last;
  }
}

```

図8 画像データを呼び出すプログラム

画像はBase 64エンコードされた文字列の状態ではキャッシュされているので、一覧表示に必要な画像データを高速に検索し、クライアントへ返送するHTML内に対象となる画像データを埋め込んで転送することができる。バイナリ形式でHTMLと別にクライアントヘッダを送る方式と比べてサーバへのクライアントからの接続回数を減らし、Perlプログラムの起動回数を抑制することで、サーバの負荷を低減できる。

キャッシュファイルの書き込みでは、複数のプログラムから同時に行われるとファイルの中身が破損するので、複数のプログラムが同時に同一のファイルへ書き込むことを防ぐ排他制御を行った。

排他制御としては一般的なflock関数を使用した。一般的なサーバ用OSではサポートされている関数である。プログラムは排他権限を得るまで1秒ごとに試行し、30秒間試行し続けても排他権限を得られなかった場合は、プログラム終了となる。

(3) 資料配信アプリ

このアプリは、教師が指定する写真・ビデオ・PDF文書などの資料をクライアントのタブレット端末から閲覧するためのものである。写真集・解説ビデオ・図鑑などのいくつかの資料を教師が指定し、児童のタブレットから必要な資料を選択して閲覧するような用途を想定している。

資料となるデータのファイルはフォルダ単位でまとめることができ、最初の画面ではフォルダの一覧が表示される。(図9) フォルダを選択するとその内部のファイル一覧が表示される。ファイル一覧のうち、画像ファイルに関してはサムネイル画像がタイル状に並べられ、閲覧性を向上させている。

資料の画像ファイルのサムネイル画像の一覧を表示する際に、元の画像データをそのままサーバから端末へ転送すると、大量のデー

タがネットワーク上に流れ、ネットワーク全体に大きな負荷がかかるので、「写真・ビデオ撮影アプリ」と同様の手法で画像を圧縮した上でBase 64エンコードし、サムネイル画像のキャッシュとHTML内への埋め込み送信をして負荷を低減している。



図9 資料配信アプリ動作画面

6. 各OS上でのアプリの実行

iOSとAndroid上で写真・ビデオ撮影アプリを起動し、カメラ機能を用いて写真を撮影して、そのデータをサーバに保存することができた。(表1・図10・図11)

表1 アプリ起動確認OS

OS・バージョン	使用ブラウザ名
iOS 6.1.3	Safari
Android 4.0.4	ブラウザ



図10 iOS上でのアプリ実行



図11 Android上でのアプリ実行

7. 写真・ビデオ撮影アプリの実践使用

(1) 対象と授業概要

6年生理科の単元「燃焼」において、蓋を閉めた集気瓶の中で蝋燭を燃焼させたときと開放空間で蝋燭を燃焼させたときの燃焼の様子の違いを観察させる実験でアプリを使用させた。

(2) 使用機材

使用機材は、第1世代iPad mini1台。第4世代iPod touch12台。Wi-Fiアクセスポイント1台。サーバ1台。AppleTV1台。大型ディスプレイ1台であった。

(3) 手順と方法

Wi-Fiアクセスポイントに無線で教師用iPad miniと児童用iPod touchを接続した。AppleTVとサーバは有線でWi-Fiアクセスポイントと接続した。AppleTVと大型ディスプレイを接続し、教師用のiPad miniに表示した画面を大型ディスプレイで表示できるように設定した(図12)。

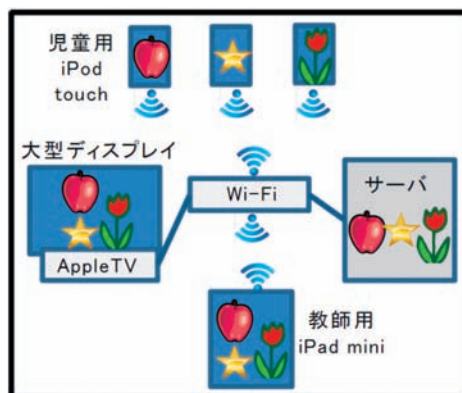


図12 使用機材間の接続方法

開発したアプリを入れたiPod touchを3～4人の班ごとに1台ずつ合計12台を配付した。

6年生はiPod touchを使用するのが初めてだったので、iPod touchの使い方とアプリの使用法の両方の説明を行った。アプリの使用を10分間許可した。この間は、数に上限なくサーバに映像を保存させた。サーバに保存された映像から教師が選んだいくつかの映像を大型ディスプレイに表示して全員で映像を共有した。6年生2学級について同様に授業を展開した。

(4) 結果

実践の結果は以下の通りであった。

- ① 6年生はiPod touchを使用するのが初めてだったが、iPod touchとアプリの使い方の説明は3分間程度で手早く済ませることができた。
- ② 合計24班が20分間に71個・63MBのファイルをサーバに保存した。平均で1分間当たり3MBのデータがサーバに保存された計算となる。サーバ負荷対策として搭載した画像圧縮とキャッシュ機能は、この実践では映像を主として扱ったので作動しなかったが、サーバが過負荷状態になることはなくアプリは正常に機能した。
- ③ カメラの向きを縦にして撮影してしまう班が多く各端末上で再生する際は問題ないが、

横長の大型ディスプレイで再生するときに画面の大きさを生かせなかった。

- ④10分間の休み時間をはさんで、2つ目の学級の授業があったが、データはサーバ上に保存されていたので、休み時間中に各端末からデータを削除するといった操作は必要なかった。
- ⑤AppleTVで映像を再生させる際に、各端末上で再生させるときよりも映像の読み込みに時間がかかり円滑ではなかった。

(5) 考察

考察は以下に項目ごとに述べる。

- ①iOSの操作は直感的で使いやすいので、使い方の説明に割く時間を少なくすることができたと考えられる。
- ②通常iOSで撮影した映像はHD画質で保存されるが、サーバ上に映像を保存する場合にはiOS側が自動的に480×360のSD画質に圧縮して送信する仕様となっている。そのため、サーバとネットワークにかかる負荷はHD画質で送信された場合と比べて低く抑えられたと推測できる。
- ③iPod touchは片手で持つときは縦向きの方が保持しやすいため、映像の撮影の際も縦向きで持ったまま撮影してしまったと考えられる。
- ④AppleTVで映像を再生する場合、教師用の端末から映像が送られるのではなく、サーバ上の映像のファイルをAppleTVがダウンロードして再生する仕様となっている。そのため、教師用端末で映像をダウンロードした後に、同一のファイルをAppleTVが改めて最初からダウンロードしなおすので再生までに時間がかかると考えられ、今後のAppleTVの機能改善を期待したい。

8. 資料配信アプリ利用の実践

(1) 対象と授業概要

4年生の総合的な学習の時間において、林

間学校の活動記録作りを行った。発表用スライドには班ごとに撮影させた林間学校中の写真を挿入させた。スライドに挿入するための画像選択や挿入順序の決定のために、iPad miniとこのアプリを用いて班ごとに保存されている画像を確認させた。

(2) 使用機材

使用した機材は、第1世代iPad mini7台。Wi-Fiアクセスポイント1台。サーバ1台。AppleTV1台。大型ディスプレイ1台であった。

(3) 手順と方法

Wi-Fiアクセスポイントに教師用1台と児童用6台の合計7台のiPad miniと、サーバを無線で接続させた。AppleTVは有線でWi-Fiアクセスポイントと接続した。AppleTVと大型ディスプレイを接続し、教師用のiPad miniに表示した画面を大型ディスプレイで表示できるように設定した(図13)。

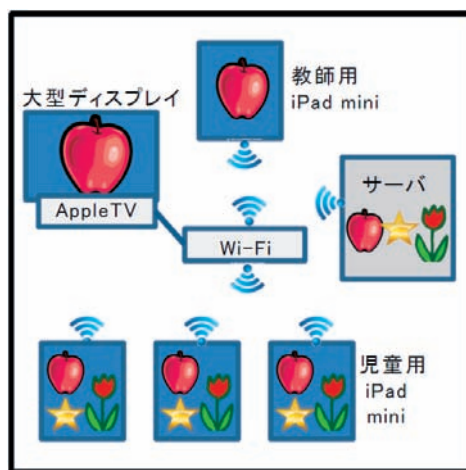


図13 使用機材間の接続方法 II

開発したアプリを入れたiPad miniを7～8人の班ごとに1台ずつ合計6台を配付した。

4年生はiPad miniを使用するのが初めてだったので、iPad miniの使い方とアプリの使用方法の両方の説明をAppleTVで教師用

端末の画面を見せながら行った。

説明終了後、およそ20分間開発したアプリを使わせて、全35フォルダに分割して保存されている合計1743枚（472MB）の写真を自由に閲覧させた（図14）。4年生2学級について同様に授業を展開した。



図14 資料配信アプリの授業における利用風景

(4) 結果

結果を以下に項目ごとにまとめる。

- ①4年生はiPad miniを使用するのが初めてだったが、iPad miniとアプリの使い方の説明は5分間程度で手早く済ませることができた。説明後、本体・アプリの操作法に関する質問は全くなかった。
- ②サムネイル画像の一覧表示の際に圧縮とキャッシュ機能を有効にしている期間は、サーバなどが過負荷状態になることはなくアプリは正常に機能した。
- ③サムネイル画像を一覧表示させる際に、4枚・21MBの容量の画像が保存されたフォルダについて、圧縮転送機能を停止したところ一覧表示までにかかる時間が顕著に増加し、児童から快適でないとの反応が出た。
- ④合計1743枚の画像データを資料として配信したが、各端末に事前の転送は必要ないので、サーバにファイルを1度転送する操作だけで配信準備を済ませることができた。
- ⑤8人の班でもサムネイル画像の一覧表示の

際に全員で確認することができていた。

(5) 考察

考察を以下に項目ごとにまとめる。

- ①iOSの操作は直感的で使いやすいので、使い方の説明に割く時間を少なくすることができたと考えられる。
- ②大量の資料を各端末へ配信する際は、このアプリのようにサーバから配信する方式にすると準備にかかる時間を短縮できるので有用な手法といえる。
- ③iPad miniはiPod touchと比べて画面が大きく、グループ学習の場面では画面が大きい端末の方がよいことが確認できた。
- ④画像圧縮機能を停止しても、サーバの負荷は余裕があったにもかかわらず、一覧表示できるまでにかかる時間が顕著に増加した原因はWi-Fiのデータ転送帯域の上限に達し、十分な速さでデータを転送できなくなったためと考えられる。

9. まとめと今後の課題

今回、4年生と6年生の児童を対象に実際にタブレット端末を使わせて実践を行った結果、利用する場面に応じて適切な大きさの端末を使用させる必要があることと、直感的に操作できるOSやアプリであれば、操作方法の説明のための時間がほとんどかからないことが明らかとなった。

そして教材用アプリ開発については、HTMLとJavaScriptという広く公開されているインターネットの中核技術を組み合わせることで、特定のOSに限定されることなく利用できるタブレット端末用アプリを開発できることが実証できた。また、サーバ上でアプリとデータを一元管理することで、タブレット端末の管理と運営を大きな負担なく円滑に進められることも明らかになった。

一方で、サーバ上でアプリとデータを一元管理する手法は、サーバとWi-Fiを用いたネッ

トワーク環境が必須であり、安定して稼働させるためにはWi-Fiとサーバにかかる負荷が許容量を超えないように、圧縮やキャッシュなどの負荷低減技術を用いる必要性和効果を確認できた。

また、今回実践したシステムはタブレット端末単体では使うことができないので、環境の整備されていない校外活動においても同様に使えるようにするには、遠隔地のWi-Fiからでもサーバに接続するための技術や屋外などで即時設置できる可搬型システムの開発、完全にWi-Fiが存在しない環境でもタブレット端末のみで使用できるオフライン実行機能を搭載したアプリの開発などが今後の課題として明らかになった。実践を通してさらにWebアプリの開発に取り組んでいきたい。

NO.1、pp.9-10 (2012)

付 記

本研究は、科学研究費補助金（平成23年度～24年度）挑戦的萌芽研究「タブレット型情報端末（iPad）を用いた授業における対話を促進するための指導方略、研究代表：今田晃一、課題番号23653306」の一部を用いて行った。

文 献

- 1) 文部科学省「教育の情報化ビジョン」
pp.2-18 (2012)
- 2) 日経パソコン「全県立校で1人1台のタブレット、佐賀県が2014年度から Windows 8かiPadか、各520台で実証実験中」教育ITソリューションEXPO講演
<http://pc.nikkeibp.co.jp/article/news/20130517/1090682/> (2013.8.11取得)
- 3) 村山大樹・大西久雄・今田晃一「教育におけるマルチタッチブック作成ソフト (iBooks Author) の可能性～iBooks を用いた実践事例とクラウド・コンピューティングとの整合性～」文教大学大学院教育学研究科『教育研究ジャーナル』VOL.5、