

研究ノート

電子書籍を用いたユビキタスラーニングシステムの構築

本池 巧

【要旨】 EPUB3形式の電子書籍では、HTML5で導入された多くの最新のWeb技術が利用可能であり、これらの機能を活用することで電子書籍はデスクトップアプリケーションと同等な機能を実現可能であり、授業における紙とタブレットの連動をより促進するものと考えられる。本稿では、EPUB3形式の電子書籍上に、授業中および時間外の受講者の学習をより効果的にサポートするとともに学習活動の追跡を可能とするユビキタスラーニングシステムの構築の取り組みについて報告する。

【キーワード】 デジタル教材、アクティブラーニング、タブレット、電子書籍、EPUB、JavaScript

1. はじめに

平成17年の中央教育審議会答申「我が国の高等教育の将来像」[1]で、21世紀は、知識・情報・技術が、政治・経済・文化などあらゆる領域における活動の基盤として重要性を増す社会（これを知識基盤社会と定義する）となると述べている。同じ頃、世界的に見た小中学生の学力レベルの低下いわゆるPISAショックは、我が国の初等・中等教育政策を「ゆとり教育」から学力向上へ転換を促す大きなきっかけとなった。今では、小中学校では学習者が主体的に学ぶアクティブラーニング型の授業が積極的に取り入れられ、一定の成果を納めつつある。

しかし、その成果は、かならずしも高校・大学に引き継がれているとはいえない。次は、高校および大学に於いて、小中学で身に着けつつある主体的な学びの習慣をさらに定着・強化する段階である。2020年度から実施される大学入試制度改革は、従来の知識偏重の入試制度を改めることで、この課題に取り組もうとするものである。

近い将来、大学には能動的学習習慣を身につけた学生が少なからず入学することになると予想さ

れる。そのような時代に備えるためにも、早急に、大学は、従来の知識伝達型の講義・実習のあり方を見直し、学習者が能動的に学ぶ形態に転換し、知識基盤社会に対応した人材育成に取り組む必要がある。

1.1 ユビキタスラーニングシステムの運用結果

アクティブラーニングによって授業内容を学習者が主体的に学べるものとしても、学びが授業の中だけにとどまっていたは意味がない。いくつかの調査結果によれば、大学生の課外学習時間が小学校高学年の半分程度という報告がある[2,3]。昨今では、経済的理由などからアルバイトの時間を増やし、学習に割く時間が確保できない学生も増えている。課外学習への取り組みを学生のやる気だけに頼るのではなく、課外学習を支援する環境作りに本格的に取り組む必要がある。

本研究ノートで扱うユビキタスラーニングシステム[4]とは、場所・時間を問わず継続的な学習を支援するシステムである。従来のeラーニングシステムと、主にWebブラウザを使って学習する点は同じである。eラーニングは映像やテキストで学習した後に確認のテストを行うなど、画

一的な知識習得とその検証に主眼が置かれているのに対して、ユビキタスラーニングシステムは、試行錯誤を通じて知識・概念を習得することを目的とする。また、eラーニングでは個々の学習項目が10～30分程度と短く、しかも学習の開始と終了などの進捗管理がシステムで管理されているのに対して、ユビキタスラーニングシステムは、学習者は継続的に学習の中断と再開を自由に行えるという特徴を持つ。

2014、2015年度の2年間、ウェブデザインの基礎を学ぶ講義におけるHTML文書の作成・デザイン手法の学習に、このシステムを活用した。講義内ではタブレット端末を用いて学習し、理解できなかった点、発展的な内容については、課外の空いた時間に、学内または自宅のPCを使って取り組む形でこのシステムを利用した。

このシステムの、学生の課外学習への影響について、学生アンケートによる課外学習時間の結果(表1)から検討する。2014年度は、このシステムの運用開始の時期であったため、学生所有のPCでの動作不具合、タブレット端末のキーボード入力などの問題で課外学習への活用がほとんど見られないが、2015年度は、これらの問題を克服し、本格的活用を開始しており、本システム導入によって課外学習を30分未満～1時間程度実施する層が、全講義科目平均を10%程度上回る結果となったことがわかる。

自己申告の課外学習の状況は信頼性が低いた

表1：ユビキタスラーニングシステムを使った講義の課外学習時間

	2014年度		2015年度	
	当該授業	全講義平均	当該授業	全講義平均
0分	50.00%	60.70%	43.90%	60.50%
30分未満	34.20%	18.00%	26.30%	16.20%
30分～1時間	7.90%	12.70%	22.80%	14.30%
1～2時間	7.90%	5.10%	7.00%	5.10%
2時間以上	0%	2.40%	0%	3.00%

め、実際のシステム利用状況から分析してみる。個人毎に、システム上のデータ使用量の合計を集計した結果のヒストグラムを図1に示す。この講義では、主にHTMLファイルを使っているため、データサイズが学習量のある程度を目安となると思われる。2014年度は平均250Kバイト付近に集中して分布している。これは、講義時間内の学習を示していると思われる。2015年度は、250Kバイト付近に分布するグループに加えて、500～1,000Kバイトの領域に幅広く分布する新たな集団が出現している。これが、課外学習に積極的に取り組む学生たちに対応していると考えられる。

ここで示したユビキタスラーニングシステムの受講生への学習活動に関する効果測定は、実際の学生の学習活動を追跡しているわけではない。このシステムが継続的学習に与える効果を正確に知るためには、受講生個人毎の細かな学習行動を知る必要がある。従来のeラーニング技術(SCORM規格)では、個人の細かな学習活動を包括的に分析することは難しいが、新しいExperience APIを導入することで可能となると思われる。

1.2 講義と実習・演習の連携

学校法人河合塾は2008年より継続的に大学の教育力に関する調査を実施し、その結果を報告している[5,6]。報告書では、大学4年間で、アクティブラーニング(AL)が学生の能動的学習を促す

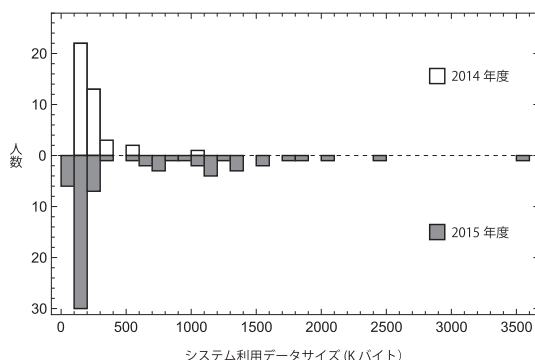


図1：ユビキタスラーニングシステムのデータ使用量

かを見るために、知識の定着・確認を目的とした「一般的なアクティブラーニング」と、知識の活用を目的としたPBL、創成授業等の「高次のアクティブラーニング」の二つに分類している。一般には、「一般的なアクティブラーニング」は、1・2年次において、講義で基礎的な知識・技術の伝達を行い、それを定着させる実習等で実施することになり、「高次のアクティブラーニング」は、3・4年次において、学んだ知識を発展させるゼミナールや演習科目等で実施していると考えられる。

今後、大学教育が知識基盤社会に適応した人材を育成するには、「講義」、「一般的なアクティブラーニング」および「高次のアクティブラーニング」を系統的に連携させたカリキュラムを構築する必要がある。

「講義」と二つのALの授業モデルにおいて用いられる教材の媒体は、一般的には、「講義」は紙が中心であり、「一般的なアクティブラーニング」は紙に加えて映像やPCを使ったマルチメディア教材が利用され、「高次のアクティブラーニング」では、さらに高度な制作・分析などに用いられる専門的なアプリケーションが使われている。赤堀による紙、タブレット、PCの特性分析によると、紙は決まった内容の記憶・理解に有効で、自分の意見を述べ発展的に考えるにはタブレットが適しているという結果が報告されている[7]。紙とタブレットは相補的な特性を持っており、組み合わせて使用すると効果的と結論している。三つの授業モデルの連携を深めるには、互い

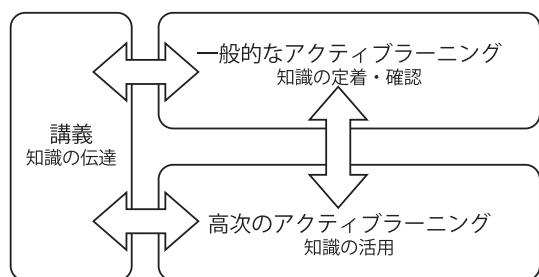


図2：講義、一般的なアクティブラーニングと高次のアクティブラーニング

の教材の相互運用が必要と考える。

今まで開発を進めてきた Web またはモバイルデバイス向けアプリをベースとしたユビキタスラーニングシステム[4,8]は、「一般的なアクティブラーニング」と「高次のアクティブラーニング」の連携を目的としたものであった。

本研究ノートでは、講義と二つのアクティブラーニングをつなぐものとして、従来の電子書籍に、アクティブラーニングで利用可能なように高度なインタラクティブ機能を追加した「インタラクティブな電子書籍」を開発し、電子書籍に学習者の学習活動を記録することで、個々の学習者の支援するシステム構築に向けた技術検証を行う。

2. 電子書籍について

紙に印刷された文書をデジタル化して保存する考え方は、HTMLの基となったSGML(Standard Generalized Markup Language)など古くからある。SGMLは特許文書やマニュアルなど長期間活用・保存される文書を電子化して管理・運用しやすくするために考えられた。1993年に、Adobe社は、紙の代わりとなる電子データPDF(Portable Document Format)を公開した。SGMLは文書の情報交換のための再利用・編集可能性を重視しているが、PDFは紙の印刷と同等の表示再現性を重視している。当初、PCやネットワークの処理能力や記録メディアの容量の制約などから、電子化された書籍・雑誌等は、一般には普及はしなかった。CD-ROMやWWWの普及に伴い、一般的なものとなり、2010年代になると、Amazon.comやiTunes Storeなどの大手のEコマースサイトで大衆全般に流通するようになった。

電子書籍を流通方式で分類すると、データをダウンロードして専用ソフトを使って閲覧するダウンロード型とブラウザを使ってオンラインで閲覧するストリーミング型の二つの形態がある。表示形態から分類すると、実際に閲覧するデバイスの画面サイズに応じて、文書レイアウトが変化する

リフロー型と画面サイズが変化してもレイアウトが変化しないフィックス型の二種類がある。

一般に、講義の中で電子書籍を扱う場合、表示デバイスやネットワーク環境に左右されないダウンロード型かつフィックス型のデータである必要があるが、スマートフォンなどを用いた課外学習の場合は、状況によっては、リフロー型やストリーミング型の方がよい場合もある。

2.1 規格とその特徴

現在、流通している電子書籍には様々なデータ型式が存在しているが、書籍編集ソフト、電子書籍リーダー、利用可能なデバイスの普及度を考えると、PDF、EPUB、AZWの三つに絞られる。それぞれのファイル形式の特徴を整理する。

PDF (Portable Document Format)

PDFは1993年Adobe Systems社によって開発された電子文書のためのフォーマットである [9]。ハードウェアおよびソフトウェア環境によらず、オリジナルのイメージを正確に再現可能である。その特性から、フィックス型の電子書籍しか作成することはできない。データ作成は、PCで動作するワープロソフトや組版レイアウトソフトであれば出力可能であり、PDFファイルを表示するためのアプリケーションもPCおよびスマートデバイス向けに様々なものがある。編集環境および閲覧環境ともに充実している。手書き、テキスト、図形など様々な形態のグラフィックデータを注釈として付加する機能を備えている。環境は限られるが、音声、ビデオや3Dグラフィックの埋め込みも可能である。JavaScriptを用いてインタラクティブな機能を付加することもできるが、入力データの妥当性チェックなど用途は限定されている。データの配布方法としては、USBメモリ等での直接配布、組織内ファイルサーバを使った配布、公開Webサーバを使った配布など様々な形態で配布することができる。講義では専らテキストやプリントをWebで公開するために使

用される。

EPUB (Electronic Publication)

EPUBは2007年に、国際電子出版フォーラム (IDPF:International Digital Publishing Forum) が策定した電子書籍のファイルフォーマット規格である。最新の規格は2011年に確定したバージョン3.0 (EPUB3) である [10]。Apple社のiOSデバイス用の電子書籍閲覧アプリiBooksでは、EPUB3の他にEPUB3の規格に独自の機能を追加したiBooks型式も利用可能である。特徴としては、文書はHTML5、表示設定はCSS3、インタラクションはJavaScriptという現在の標準的なWebの技術を使っている点である。そのためレイアウトソフトだけでなくWebオーサリングソフトやテキストエディタなど様々な方法で作成可能である。EPUB型式のデータ作成Webコンテンツと同様、当初はリフロー型の電子書籍しか作成できなかったが、EPUB3からフィックス型にも対応している。音声やビデオの埋め込みも可能で、日本語独自の、縦書き、ルビにも対応している。HTML5のマルチメディア機能やJavaScriptが利用可能なことから高度なインタラクションを実現することができる。ただし、HTML5とJavaScriptを組み合わせたインタラクションに関してはEPUBファイルリーダーによって対応状況が異なるため、講義で配布する場合には注意が必要となる。配布方法はPDFと同様に様々な形態が可能である。

AZW (Amazon Whispernet)

Amazon Kindle専用の電子書籍のデータ型式である。Amazon.com社の独自仕様でありファイル規格は一部しか公開されていない [11]。この型式のデータを作成可能なアプリケーションは限られている。配布方法は、主にAmazon.com社のクラウドサービス経由であり、限定的である。音声、ビデオの埋め込みが可能であり、HTMLとCSSを一部利用できるが、JavaScriptが使え

ないため、インタラクティブな機能を付加することはできない。この型式は、書籍や雑誌などを Amazon で販売するためのもので、講義での利用は、市販の電子書籍を使う時に限られる。

以上、現在流通している電子書籍の代表的な三つのファイル形式について概観した。本研究の目的である、「インタラクティブな電子書籍」を実現するには、機能、作成環境、配布方法から判断して、EPUB 型式がふさわしいと結論する。なお、ハードウェア、リーダによってインタラクティブな機能への対応状況が異なっているが、本研究では、普及率および対応する機能の面で iOS デバイス (iPhone、iPod Touch、iPad) で動作する iBooks を取り扱うこととする¹⁾。

2.2 電子書籍オーサリング環境

EPUB3 の中身は HTML+CSS であると述べたが、おおまかには章 (またはページ) ごとのコンテンツを XHTML5 型式で記述し、レイアウト設定のスタイルシートや画像などの素材、インタラクションを定義する JavaScript ファイル等と書籍に関するメタデータを記述したファイルを ZIP 型式²⁾で圧縮したものである (図 3)。

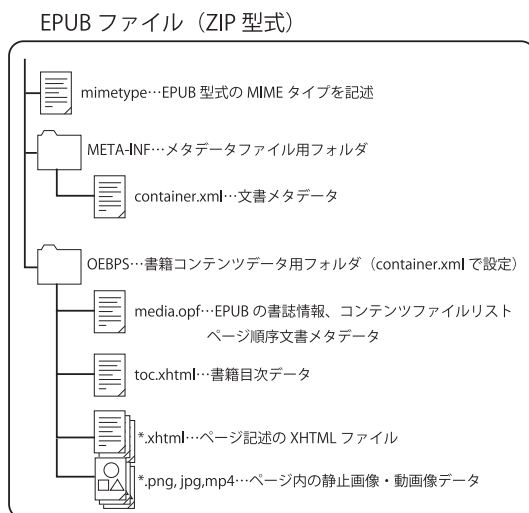


図 3 EPUB3 ファイルの構造 [10]

図 3 から分かるように、書籍コンテンツデータフォルダ内部は、Web コンテンツ同様な構造になっている。EPUB 制作環境には、印刷文書作成向けのレイアウトアプリケーション (InDesign など)、Web オーサリングアプリケーション (Dreamweaver など)、電子書籍用のオーサリングアプリケーション (iBooks Author など) がある。図 3 のように構成されたフォルダ・ファイルを EPUB3 にパッケージ化するアプリケーションを使えば、テキストエディタのみでも作成することができる。

フィックス型のインタラクティブな EPUB ファイルを作成する場合、レイアウトアプリケーションまたは電子書籍用オーサリングアプリケーションが適しているがインタラクションの作成能力が不十分である。JavaScript インタラクションを組み込む場合は、Web 用のオーサリングアプリケーションが適しているが、フィックス型のレイアウトに適していない。現状では、「インタラクティブな電子書籍」に必要な機能を持ったアプリケーションが存在しない状況であるが、本研究では、電子書籍用のオーサリングソフト iBooks Author を用いることとする。理由としては、フィックス型の書籍に対応し、実用レベルのレイアウト機能を備えていること、HTML ウィジェットという機能を使って JavaScript を使ったインタラクティブなコンテンツの埋め込みが可能であり、動作検証を目的とするために十分な機能を持っていると判断したからである。

2.3 電子書籍の機能拡張

HTML5 は HTML4 に比べて、大幅な機能拡張が施されており、すべてではないが多くの機能が EPUB 型式の電子書籍で利用可能である。以下に、教材作成で重要と思われる機能についてまとめる。

Canvas API

Canvas 要素内に、基本図形、自由曲線などを

JavaScriptで描画することを可能とする。従来、flashplayerなどのプラグインを使って実現していたアニメーションがブラウザ単体でも可能となっている。これを応用したJavaScriptライブラリも多数公開されている。代表的なものとして、グラフやチャート作成のd3.jsや、ビジュアルプログラミング環境Processingと同等な機能を、ブラウザ上に実現するp5.jsなどがあげられる。

Video/Audio 要素

HTML4まではflashplayerを使う必要があった音声やビデオ再生をブラウザのみで可能とする。さらに、WebAudio APIを使用することで、サウンドデータのリアルタイム処理や、Text-to-Speechが可能となる。

Web Storage

従来はクライアント側にデータを保存する場合はCookieを使用していた。Web Storage[12]は、Cookieに比べて、サイズ制限がない、サーバに送信されない、有効期限がない、JavaScriptのデータを直接保存できるなどの特徴をもち、Cookieよりも柔軟に情報を保存できる。電子書籍の場合、ユーザの学習情報を記録するためには欠かせない機能である。

WebSocket API/WebRTC API

WebSocketとは、ブラウザとサーバ間で直接通信を実現する機能である。従来クライアントはWebサーバとしか通信出来なかったが、WebSocketを使えば様々なシステムとの直接通信を可能とする。WebRTCは、サーバを介さず、クライアント同士での直接通信を可能とする。アクティブラーニングでよく活用されるクリッカーシステムなどは、WebSocketやWebRTCを使えば容易に実現可能である。

JavaScript ライブラリ

高度なインタラクティブなJavaScriptで初め

から制作することは時間と手間がかかり、本来の教材作成に割く時間が減ることとなる。Web製作用に様々な使いやすいJavaScriptライブラリ(代表はjQuery)が公開されており、活用することで教材作成の時間と時間の大幅削減が可能となる。

なお、EPUBリーダのなかには、HTML5に未対応の古いEPUB2規格までしかサポートしていないものや、EPUB3対応でも、使えるHTML5の機能に制約があるものがある。スマートデバイスの場合、PCに比べ、処理能力、ネットワーク速度や扱えるデータサイズに関する制約が大きいため、講義で実際に使用する前に、様々な環境での動作確認が必要である。

3. 高度なインタラクティブ電子書籍の作成

電子書籍を講義等で使用する場合、受講生が使用するデバイスでの表示と教員がプロジェクト等での表示において、内容やページ番号等を一致させなければならない。そのため、フィックス型の書籍となる。また、インタラクティブな要素と文章を連携させるためにもフィックス型が適している。フィックス型かつインタラクティブな電子書籍を作成する方法を説明する。

3.1 電子書籍の機能拡張の仕組み

iBooks Authorは、Apple社の文書作成アプリケーションPagesやプレゼンテーションアプリケーションKeynoteとよく似た画面構成で、フィックス型の電子書籍作成を前提としている。

iBooks Authorの特徴は、ワープロと同じ操作で電子書籍を作成できるだけでなく、ウィジェットと呼ばれるパーツを使ってインタラクティブな要素の追加を簡単にすることで、フィックス型の電子書籍におけるレイアウト作業と、インタラクティブな機能の連携を可能としている。図5にiBooks Authorで使用できるウィジェットの一覧を示す：

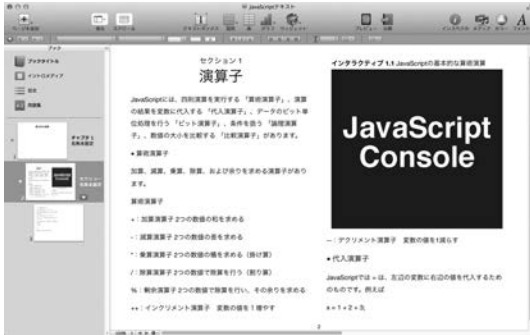


図 4：iBooks Author の画面

- ・ギャラリー：複数の写真をスライドショーで表示
- ・メディア：音声・ビデオの埋め込み
- ・練習問題：選択肢型式の問題作成
- ・Keynote：Keynote で作成したスライドの読み込み
- ・インタラクティブ：画像に対するインタラクティブな説明
- ・3D：3D 図形の埋め込み表示
- ・スクロールサイドバー：長い文章をスクロールバー付きテキストボックスで表示
- ・ポップオーバー：画像クリックでポップアップテキストを表示
- ・HTML：JavaScript による HTML5 形式のインタラクティブ要素の埋め込み

3.2 独自のウィジェットの構築

ウィジェットとは、元々は Mac OS X 10.4 以後で、カレンダー、電卓など小さなアプリケーションをデスクトップ上で実行する Dashboard という環境の中で実行される個々のアプリケーションのことである。ウィジェットは、HTML+CSS でユーザインターフェースを、JavaScript でアプリケーションの動作を記述する。iBooks Author では、HTML ウィジェットを使って Dashboard 用のウィジェットの書籍本文への埋め込みを可能としている。

ウィジェットの開発には、以下の三つの方法が



図 5：iBooks Author のウィジェット

ある。

1. インタラクティブな HTML5 コンテンツ開発用アプリケーション (iAd Producer、Dashcode、Hype、MotionComposer など) を用いる方法
2. テキストエディタを使う方法
3. 独自ウィジェット作成用の Web サービス (Bookry など) を使用する方法

HTML5 は新しい技術が続々と開発されており、それらを最大限に活用するには、2 番目のテキストエディタを使って、HTML ファイル、CSS ファイル、JavaScript ファイルを作成する手法が最適である。

HTML ウィジェット自体は名前に拡張子、

wdgt がついたフォルダで、図6に示すように、その中に、Info.plist、Default.png、ウィジェット内のHTMLコンテンツが含まれている [13]。

Info.plist ファイルは以下のように、ウィジェットの幅、高さ、メインHTML ファイルを設定する。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Width</key>
  <integer>ウィジェットの幅</integer>
  <key>Height</key>
  <integer>ウィジェットの高さ</integer>
  <key>MainHTML</key>
  <string>メイン HTML ファイル</string>
  <key>CFBundleDevelopmentRegion</key>
  <string>English</string>
  <key>CFBundleIdentifier</key>
  <string>com.trevorburnham.widgetboilerplate</string>
  <key>CFBundleName</key>
  <string>iBooks HTML Widget Boilerplate</string>
</dict>
</plist>
```

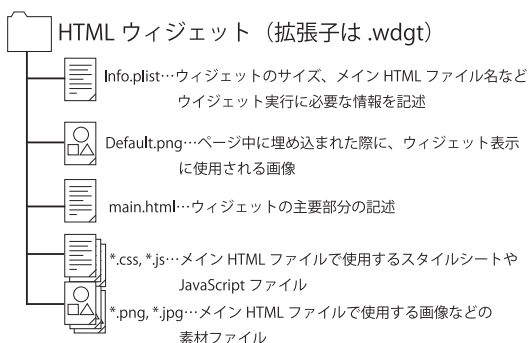


図6: HTML ウィジェットの構造

メインのHTMLコンテンツは、ウィジェット独自の設定はなく、一般的なHTML制作の手法で制作可能である。

3.3 ユビキタスラーニングシステムとして必要な機能

本研究では、まず、現在運用中のWebベースまたは専用アプリとして構築してきたユビキタスラーニングシステム [4,8] と同等な機能を持つ電子書籍に取り組んでいる。従来のユビキタスラーニングシステムは、講義で学んだことの確認・応用を目的としており、一回の学習が15分程度必要とするものであった。今回構築している電子書籍ベースのものは、知識の定着・確認を目的としており、一回の学習が数分程度で終わる程度の短いものとしている。

基本的な構成を図7に示す。書籍は学ぶ内容を解説する文章があり、その文章の終わりにHTMLウィジェットを使った簡単な確認作業を行う。今までのユビキタスラーニングシステムを運用した経験より、講義で全員が無線LANを使って接続した場合、サーバとの通信が安定せず学習がうまくできないというトラブルが一定の頻度で発生した (1%程度)。今回は、サーバとの接続は学習情報の送受信のみで使用し、それ以外は電子書籍内で処理をすることとする。

このようなシステムを構築するためには、学習者の学習情報を書籍内への保存と、書籍内でのプログラムの実行ができなければならない。

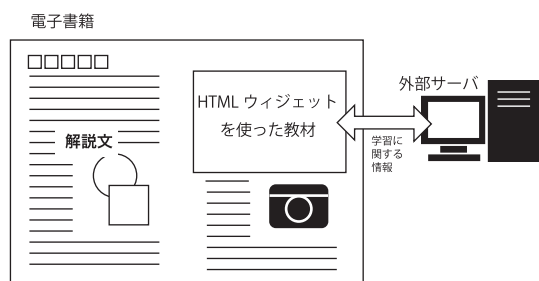


図7: 電子書籍を用いたユビキタスラーニングシステム

学習情報の保存

学習情報の保存方法として、Web Storage API の Local Storage が有効と思われる。実際に、この API を使って学習者の ID を電子書籍に保存できるかどうかの検証を行った。検証用のウィジェットは、フォームから入力した文字列を Local Storage に保存する。Local Storage は Key-Value 方式のデータで、読み出しは getItem (キー名)、書き込みは setItem (キー名、値) で行う。実際に、フォームに入力された情報を Local Storage に保存する処理を以下に示す。

```
$(function(){
  // Check browser support
  if (typeof(Storage) !== "undefined") {
    // Store
    if(localStorage.getItem("myID")
      === null){
      localStorage.setItem("myID", "Alice");
    }
    $('#userID').val(localStorage.
      getItem("myID"));
  } else {
    document.getElementById("result").
    innerHTML = "Sorry, your browser does
    not support Web Storage...";
  }
  $('#btn').click(function(){
    localStorage.setItem("myID",
    $('#userID').val());
  });
});
```

検証用のウィジェットの実行状況を図 8 に示す。まずはウィジェットの初期画面 (Default.png の内容) が表示される (図 8(a))。ウィジェットをタップすると、入力フォームが現れ、ID を入力しボタンを押せば登録完了となる (図 8(b))、その後は ID がウィジェット内に常に表示される。

Local Storage に保存されたデータは、ページの切り替え、書籍リーダの終了に関わらず保存される。一度設定した情報は明示的に消去しないと消えないため、個人情報の管理が実際の運用では課題となる。また、Local Storage の記憶領域は書籍全体で共通しており、一冊あたり最大 5MB



(a)



(b)



(c)

図 8：電子書籍にユーザ ID を保存

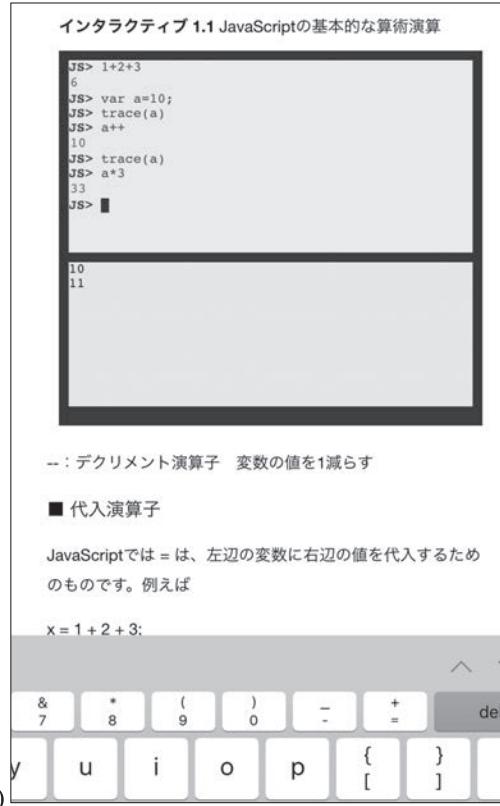
の容量制限がある。分量の大きい教材では、複数冊に分割したり、cloud システムに記録するなど、学習情報のデータ管理が必要となる。

書籍内でのプログラムの実行

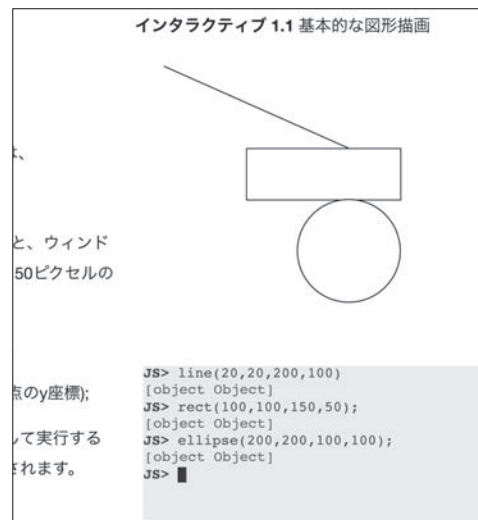
JavaScript には、文字列に格納された命令を動的に評価する eval 関数がある。この関数を用いて、入力された JavaScript の命令を実行し、その結果を表示するウィジェットを作成し、実際に、電子書籍内で入力したプログラムの実行が可能かどうかの検証を行った。ウィジェットでは、エラー処理を含め以下の関数を使って入力された JavaScript の命令の実行を行った³⁾。

```
commandHandle:function(line){
    var geval = eval;
    try {
        var ret = geval(line);
        if (typeof ret != 'undefined')
            return ret.toString();
        else
            return true;
    }
    catch (e) { return e.toString();
    }
}
```

なお、プログラムの入力とその結果表示には、jQuery [14] をベースとするライブラリ jquery-console [15] を使用した。電子書籍内で入力した JavaScript の命令を、書籍内で実行している様子を図 9(a) に示す。ビジュアルプログラミング環境 Processing と同等な描画機能を提供する JavaScript ライブラリ p5.js を組み合わせれば、図 9(b) に示すように、電子書籍内で、コマンドを使ったグラフィック処理も可能である。さらに、Web Audio を使ったりリアルタイムでの音響合成や、Web Socket を使った Web サーバなど、PC 上で行っていた実習の多くが電子書籍



(a)



(b)

図 9：電子書籍内でのプログラムの実行

上で実際に操作しながら学ぶことができることがあきらかとなった。

4. まとめ

現在の電子書籍リーダーは Web ブラウザとほぼ同等の機能を有しており、ある程度の制約はあるが、デスクトップアプリケーションとほぼ同程度の機能を書籍内に埋め込むことが可能であることが実証できた。

従来の e ラーニングでは、PC を用いた学習が一般的である。しかし、この形態は、PC に電源を入れ、ログインし、最終的にアプリケーションを立ち上げるまでに数分を要するため、(宿題などの) 学習開始に動機付けを必要とする。インタラクティブな電子書籍では、思い立った時にすぐに学習を開始することができ、学習者の主体的な学習活動を促すことができると思われる。今後は、実際の講義等で運用し、その効果、課題等について分析を行い、インタラクティブな電子書籍の長所・短所、どのような構成の教材が効果的であるかを明らかにする必要がある。

最後に、今後、電子書籍を用いた教材での学習行動をどのように記録するかについて考えてみたい。

現在、大学教育において進行中の、知識伝達型教育から学習者の主体的な学びへの転換と同様なパラダイムシフトが e ラーニングでも起きている。

1990 年後半ごろから、PC を用いた CBT (Computer Based Training) の導入が進み、インターネットの普及に伴って Web サーバに教材を置きブラウザで学習する WBT (Web Based Training) へと進化していった。中核となるサーバは LMS (Learning Management System) と呼ばれ、教材の配信、学習に関する情報管理などを行う。WBT の普及にともなって米国の ADL (Advanced Distributed Learning) により LMS 間での教材の相互利用を促進するための e ラーニング規格 SCORM (Sharable Content Object Reference Model) が定義された [16]。SCORM では、

図 10 に示すように、学習者は、LMS に配置された学習プログラムの構造や進捗管理に従って配信される教材 (SCO: Sharable Contents Object) を Web ブラウザで学習し、学習結果は随時 LMS へと配信される構造になっている。

SCORM では、学習の順序や内容はすでに決められており、学習内容を自由に選択することもできない知識伝達型のシステムである。学習者が主体的に学ぶアクティブラーニングの中での SCORM 規格に基づく e ラーニングシステムの活用は難しい面が多いと思われる。

アクティブラーニングにおいて必要な e ラーニングシステムは、学習者の様々な学習活動を記録・追跡できるものでなければならない。ADL では、学習者の多様な学習活動に対応した Experience API という規格を 2013 年に定義・公開している [17]。SCORM では、LMS 内の教材の学習開始から完了の記録まで細かく記録するのに対し、Experience API では、学習者の学習体験のみを記録する (図 11)。実際には、体験を、誰 (Actor)、体験したこと (Verb)、どのような活動において (Activity) の三つから構成されるステートメントとして記述し、それを LRS (Learning Record Store) に記録するという単純な構成になっている。

Experience API は JavaScript で構築されており、今回構築したインタラクティブな電子書籍に

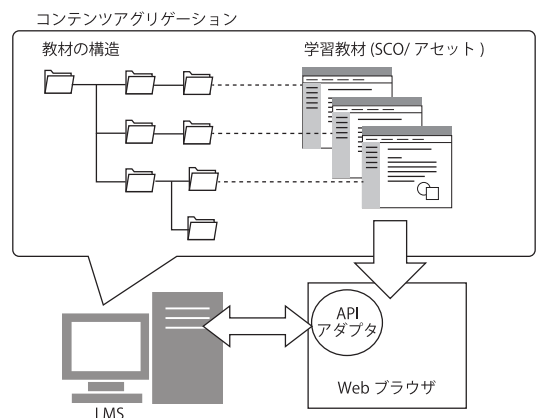


図 10：SCORM の構造

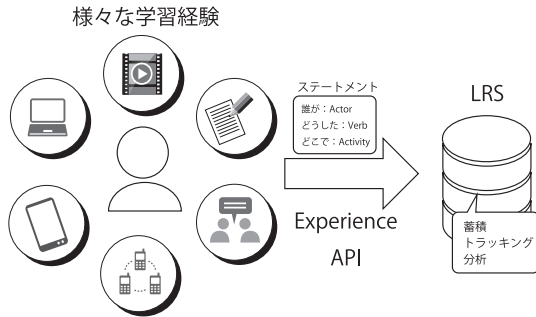


図 11 : Experience API の構造

組み込むことが可能である。しかも、SCORM と違って、ネットワークに接続されていない状況でも利用可能であるため、Experience API を組み込んだ電子書籍は学習体験をより幅広く追跡することが可能である。今後は、Experience API を使って学習活動を追跡できるユビキタスラーニングシステムを構築し、アクティブラーニングで活用することで、ユビキタスラーニングシステムの有効性・課題について明らかにする予定である。

注

- 1) 筆者の環境で確認した範囲では、埋め込みビデオの再生ができたのは iBooks だけであった。
- 2) 中身は ZIP 型式データだが、拡張子は .epub を使う。
- 3) try 内での実行時に、関数呼び出しのコンテキストにアクセスできるようにするため eval 関数を変数に代入している。これによって、変数宣言命令を動的に実行可能とする。

参考文献

[1] 文部科学省中央教育審議会, “我が国の高等教育の将来像 (答申)”, http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/05013101.htm, 2005.

[2] 内閣府, “日本の青少年の生活と意識 (第2

回調査)”,

<http://www8.cao.go.jp/youth/kenkyu/seikatu2/pdf/0-1.html>, 2001.

[3] 文部科学省, “学生の学修時間の現状”, http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo4/siryu/attach/_icsFiles/afldfile/2012/07/27/1323908_2.pdf, 2012.

[4] 本池 巧, “ユビキタスラーニングシステムの構築”, 「メディアと情報資源」20(1), 2013.

[5] 河合塾, “アクティブラーニングでなぜ学生が成長するのか—経済系・工学系の全国大学調査からみえてきたこと”, 東信堂, 2011.

[6] 河合塾, “「深い学び」につながるアクティブラーニング—全国大学の学科調査報告とカリキュラム設計の課題”, 東信堂, 2013.

[7] 赤堀侃司, “タブレットは紙に勝てるのか”, Jam House, 2014.

[8] 本池巧, “アクティブ・ラーニングを支援するインタラクティブなデジタル教材開発に向けて”, 「メディアと情報資源」21(2), 2014.

[9] ISO, “ISO 32000-1: 2008 Document management – Portable document format – Part 1: PDF 1.7”, http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000_2008.pdf, 2008.

[10] International Digital Publishing Forum, “EPUB 3.0.1”, <http://idpf.org/epub/301>, 2014.

[11] Amazon.com, “Kindle Format 8”, <https://kdp.amazon.com/help?topicId=A3IZVEA3T648O2>, 2011.

[12] W3C, “Web Storage”, <http://www.w3.org/TR/webstorage/>, 2015.

[13] Apple Inc., “About HTML widget creation in iBooks Author”, <https://support.apple.com/en-us/HT204433>, 2015.

[14] jQuery, <https://jquery.com>.

[15] jquery-console,

- <https://github.com/chrisdone/jquery-console>.
- [16] ADL, “SCORM 2004 4th Edition”,
<http://adlnet.gov/adl-research/scorm/scorm-2004-4th-edition/>
- [17] ADL, “Experience API”,
<https://github.com/adlnet/xAPI-Spec/blob/master/xAPI.md>, 2015.

Development of Ubiquitous Learning System on Digital Books

By Takumi Motoike

[Abstract] By making use of modern WEB technologies introduced by HTML5, EPUB3 digital books realize same functionalities as desktop applications, and promote the effectiveness of combination of printed materials and tablets. In this paper, we presents development towards a ubiquitous learning system based on EPUB3 digital books, which will effectively support and track the student's learning activities in coursework and homework.

[Key words] digital learning material, active learning, tablet, ebook, EPUB, JavaScript