

# Web 技術を活用した柔軟で効果的な遠隔講義向け筆記共有システムの開発

代表研究者 三浦元喜 九州工業大学 大学院工学研究院 基礎科学研究系 准教授

## 1 研究調査の要旨

本研究調査では、Web 技術を活用した柔軟で効果的な遠隔講義向け筆記共有システムの研究として、主に以下の2点の技術開発を行った：(1) デジタルペンと WebSocket 技術を用いて、遠隔地における紙への筆記をリアルタイムに Web ブラウザ上で共有・閲覧できるシステムの開発 (2) アニメーションを伴って再生可能な時系列付きの筆記情報を、永続的に Web アクセス可能にしたり、共有を促したりするための、Web ベースの筆記閲覧・描画環境の構築。

また後者の (2) に関連して、学生が筆記情報を含む Web コンテンツを簡単かつ手軽に作成・編集・共有できることを目的として、Web ブラウザ上で動作するエディタを利用した Web 開発環境 Sweetie を構築した。

## 2 研究背景と目的

教育の情報化が浸透し、一般的な教室においてタブレットやノート PC、スマートフォンなどのデジタル機器を学習者が使用する機会が増えている。これらのデジタル機器は、ディスプレイに教材や講義資料を表示することができたり、教師からのフィードバックを提示することができるため有用性が高い。こうしたデジタル機器やデバイスは、レスポンスアナライザのように選択的な応答を返すのには適しているが、従来の授業で主に行われてきた紙に書く／描くことによる学習活動との親和性はそれほど高くない。

紙に書く／描く行為は、頭のなかにある考えやアイデアを手軽に表出できるという特徴がある。また、その行為によって得られた時系列の筆記情報は、その行為（筆記）を行った人の思考の流れや、それが書かれたときの状況（コンテキスト）を含んでいる。記録した時系列付きの筆記情報を再生しながら参照することによって、行為者の思考の流れやコンテキストを、参照する人がある程度推測することができる。よって、時系列が含まれている手書き筆記情報を共有・流通させることは、コミュニケーションを円滑にしたり、他者の考えを理解するうえで有益であると考えられる。

教育が行われる環境において、とくに同期型遠隔講義（ストリーミング中継によるリアルタイム型の講義）では、講師が受講者の状況や理解度を把握しにくかったり、受講者同士が遠隔地の状況を認識しにくいといった、コミュニケーションやアウェアネスに関する問題が発生しやすい。通常の同期型遠隔講義では、板書やスライドを含む講師側の映像と音声を遠隔地の受講者側に送ることが最低限の要件として行われているが、コミュニケーションやアウェアネスに関する問題を緩和するために遠隔地の受講者の映像を講師側に送信したり、講師側にいる受講者の映像を遠隔地に送信したりすることも行われてきた。ただし、講師にとっては、映像だけでは遠隔地の受講者の状況を十分に把握することは困難であるため、どうしても一方的な知識伝達型の講義になりやすい[1]。

本研究を開始する前に、我々は遠隔講義において講師が受講者の状況を把握しやすくするため、同期型遠隔講義に対応するデジタルペン筆記共有システム WebATN を構築してきた[2]。これは、我々が以前構築したデジタルペン筆記による学習者間相互作用を促すシステム AirTransNote[3,4]を改良し、受講者が紙に書いた筆記を Web サーバに自動的に蓄積し、Web ページとして閲覧できるようにしたものである。図 1 に、WebATN システムの筆記一覧画面を示す。受講者が書いた文字や図といった手書き筆記がサムネイル表示されており、クリックすると拡大して表示する機能を備えている。この仕組みにより、たとえば出席代わりのレポートやアンケート、授業の感想、ミニツッペーパーを受講者に書いてもらおうと、それらが集約され、講義時間内に詳細な筆記情報として参照できる。これにより受講者がどこまで講義内容を理解しているかを、講師が把握することが容易になり、講義進度の調整や、個別の受講者に対する対応が行いやすくなる。

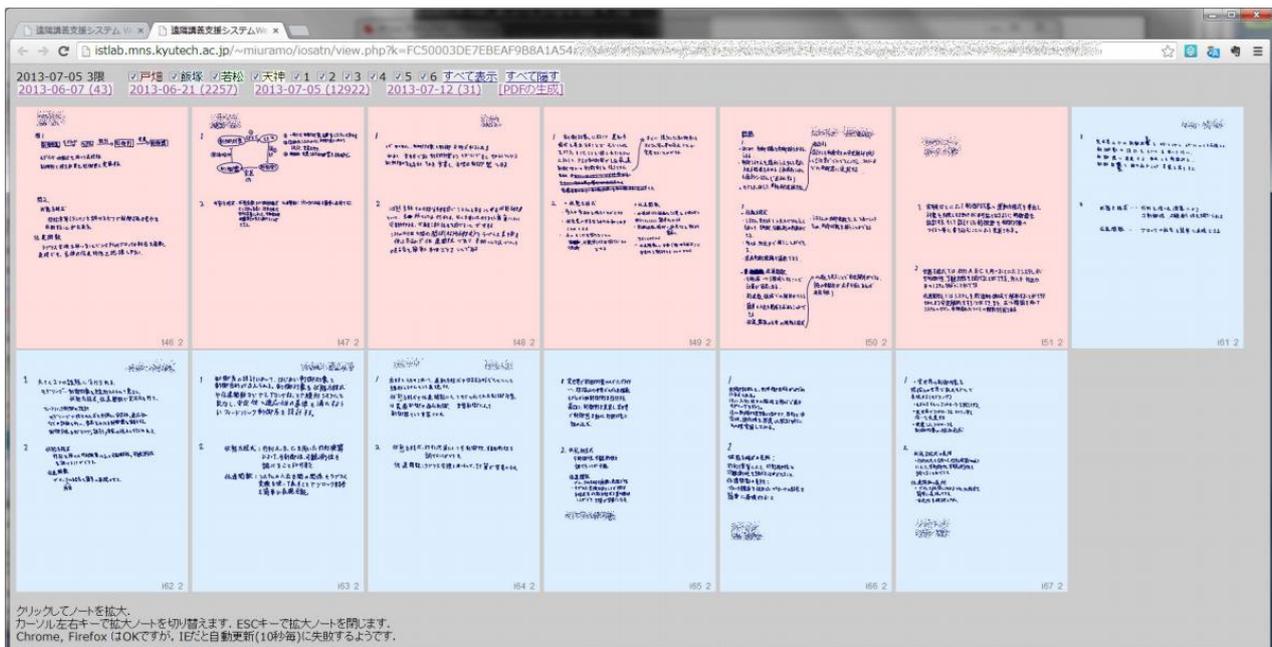


図 1 : WebATN システムの筆記一覧画面

### 3 WebATN の利点と問題点

WebATN のように、紙への筆記を遠隔地間で共有するシステムを、HTML5 をはじめとする 標準的な Web 技術を活用して構築することの利点として、筆記を参照するデバイスの自由度が高まり、運用しやすいことがあげられる。Web ブラウザさえあれば利用できるため、PC をはじめ、タブレット (iOS / Android) やスマートフォンでも動作させることができる。専用のソフトウェアやアプリケーションとして提供する場合、それを導入するための手間が必要になったり、対応 OS が使えるハードウェアを準備しないと使えないといった制約が発生することがある。教育現場においては、機材の導入時期や機材選定の判断、予算の制約によって、様々な機種や OS が混在するといったことも考えられる。Web ブラウザ上で閲覧可能な Web ページとして提供することで、OS のアップグレードに伴って、アプリケーションの更新をする手間なども省くことができる。

WebATN の問題点 (主に使用しているデジタルペンの制約) として、以下が挙げられる。

- 筆記の更新がリアルタイムに行われない。
- 送信チェックボックスをタップして送信しているあいだは筆記が行えない。

我々が従来利用してきたデジタルペン (小型省電力 Linux マシンを受信端末として Bluetooth 通信により集約する方法) では、受講者が送信チェックボックスをペン先でタップすると筆記の送信を開始し、送信が終わるまで数秒間待つ必要があった。そのため、受講者が送信を忘れてしまうと筆記が収集できないという問題があった。また、受講者が頻繁に送信操作を行いたくても、送信中は筆記が行えないため、思考の流れが中断されてしまうという問題があった。本来は、受講者が意識しなくても、自動的に送信が行われ、かつその送信筆記が Web ページ上にリアルタイムに反映されることが望ましい。

我々は筆記共有による遠隔講義中のアウェアネス伝達の効果を最大限に高めるため、ストリーミングに対応したデジタルペンを使用し、かつ WebSocket を用いてサーバ側の更新をリアルタイムにクライアントに通知する動作を実現する。これにより、従来必要であった、筆記を一旦中断して送信作業を行うための意

識の切り替えが不要になるとともに、書いている状況が即座に Web ページに反映されるため、専用ソフトウェアに近いインタフェースを Web ブラウザ上で提供できるようになった。

## 4 リアルタイム性の高い筆記の Web 共有

リアルタイム性の高い筆記の Web 共有を実現する仕組みについて、(1)デジタルペンデータの入力(2)リアルタイムに Web ページを反映する方法 に分けて説明する。

### 4.1 デジタルペンデータの入力

紙への時系列筆記情報を、筆記を行っている最中を含めて、リアルタイムに取得するため、ストリーミング技術に対応したデジタルペンと、大日本印刷が構築しているデジタルペンソリューション研究用 IM 開発キットを利用した。この IM 開発キットに含まれているアプリケーション開発用 SDK (Streaming SDK for DSPD) を用いて C#アプリケーションを作成すると、ペンデータ送受信ツール DSPD (DNP ストリーミングペンドライバ) を用いて、PC に接続した Bluetooth USB ドングルを介してデジタルペン (日立マクセル DP-601) の筆記データをリアルタイム受信することができる。受信可能なデータはイベントとして即座にコールバック関数に渡されるため、以下のデータ (表 1) をリアルタイムに取得できる。

表 1 : リアルタイム受信可能なイベントと、得られるデータの種類

イベント\データの種類	ペンの ID	座標	紙の ID	筆記圧
ペンダウン	○			
新しい座標の入力 (ペンダウン中は、座標 が変化していなくても連続 的に入力)	○	○	○	○
ペンアップ	○			

使用にあたっては、Bluetooth USB ドングルとデジタルペンを 1 対 1 でペアリングしておく必要があるが、一旦ペアリングしておけば、どの PC に USB ドングルを接続しても、すぐに筆記情報を転送可能な状態となるため、PC を変更することは容易である。

デジタルペンからの筆記を受信するアプリケーションは C#で実装する必要があるため、WindowsOS に限定される。筆記を処理するプラットフォームの制限を緩めるため、C#のプログラムでは UDP (User Datagram Protocol) で筆記データを送信する仕組み (筆記中継システム) を構築した。

### 4.2 リアルタイムに Web ページを反映する方法

従来開発してきた WebATN (Web 版 AirTransNote) [2]では、新しいデジタルペン筆記の確認にポーリングを利用していた。そのため、筆記の更新頻度を高めるためにはポーリングの回数を増やす必要があるが、回数を増やすとサーバの負荷が高まってしまうため、効率が悪いという問題があった。

そこで我々は、WebSocket 技術によりサーバ側の変更をクライアントに即座に反映することができ、リアルタイム性の高い筆記共有 Web アプリケーションを実現できるフレームワーク Meteor[5]を利用した。Meteor は Javascript で記述された Web サーバ基盤 Node.js に基づいて構築されている Web アプリケーションフレームワークである。MongoDB データベースが組み込まれており、サーバ側のデータベースの情報が変更されたときに WebSocket を用いてクライアント側の表示を自動的に更新する仕組みが備わっている。また HTML のテンプレート機能が同梱されており、シンプルな記法で Web アプリケーションを記述できるという特徴がある。

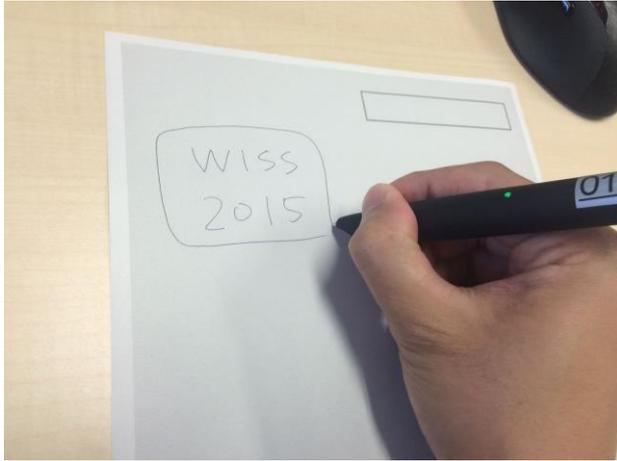


図2：ストリーミング対応デジタルペンによる記入

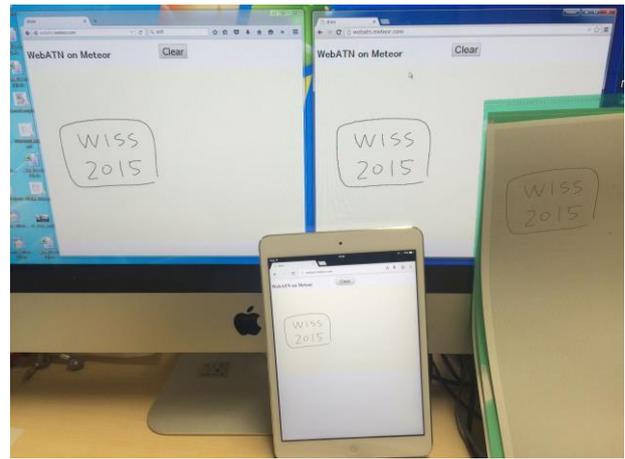


図3：筆記内容を複数のブラウザで閲覧

**Meteor** では、クライアントがサーバに接続し、**MongoDB** にデータを追加したり、サーバがクライアントに最新のデータを配信・提供するための **DDP (Distributed Data Protocol)** と呼ばれるプロトコルを規定している。我々は、筆記中継システムからの筆記情報をリアルタイムに **MongoDB** に格納するため、UDP で受信した筆記情報を、**DDP** で指定したサーバに送信するアプリケーション（筆記格納システム）を **Java** で開発した。また、**Meteor** フレームワーク上で、**MongoDB** に格納された筆記情報（座標）を **Web** ページ上でリアルタイムに更新表示するプログラム（筆記配信システム）を構築した。

筆記格納システムは、デジタルペンの座標が更新されるたびに、**DDP** を用いてサーバに接続し、新しい座標を **MongoDB** に追加する。すると、**Meteor** がそれを検知し、更新用の **DDP** メッセージを接続中の **Web** クライアントに通知する。これにより、デジタルペンで書いている内容（図2）が、接続しているクライアントの **Web** ブラウザに即座に表示される（図3）。ストロークの完了（ペンアップ）を待たずに筆記が更新されるため、遠隔から筆記を参照しているときにも、どこに何を追記しているのかがわかりやすい。**Meteor** は標準的な **Web** アプリケーションとして動作し、多くのモダンな **Web** ブラウザに対応している。そのため、**PC** に限らず、**AndroidOS** や **iOS** を搭載したタブレットやスマートフォンからも接続し、筆記をリアルタイムに確認することが可能となる。

今回開発した仕組みは、手書き筆記を遠隔地間でリアルタイムに共有する手段としては基礎的な部分に限られているが、オープンな技術を中心に構成されており、柔軟性および汎用性は高い。現時点では1台の **PC** に筆記中継・格納・配信システムを動作させ、10本のペンによる同時筆記ができることを確認している。今後表示方法の改良やシステムの最適化を行うことで、より多人数での同期型遠隔講義に対応できるようにしていきたいと考えている。

## 5 動的な筆記情報の **Web** 共有を促すための筆記閲覧・描画環境の構築

これまでに述べた、時系列情報を含む手書き筆記をリアルタイムに共有するシステムは、**Meteor** フレームワークを使用して構築してきた。**Meteor** は手軽にサーバを起動することができる点で優れたフレームワークであるが、永続的で大規模なアプリケーションには対応しづらいという問題があった。時系列情報を含む手書き筆記を、永続的に **Web** アクセス可能にするため、我々はアニメーションを伴って筆記を再生できる閲覧環境と、そのようなサイトを手軽に構築できる **Web** アプリケーションを作成した。

最初に、時系列筆記データを図4のようなシンプルな **JSON** 形式で準備しておく。1ストローク分の筆記データは、筆記の座標  $x,y$  を含む配列データが、筆記の点の数だけ含まれる配列データとして構成している。データファイル全体としては、ストロークの数だけ、ストロークデータを含む、1つの配列データとなって

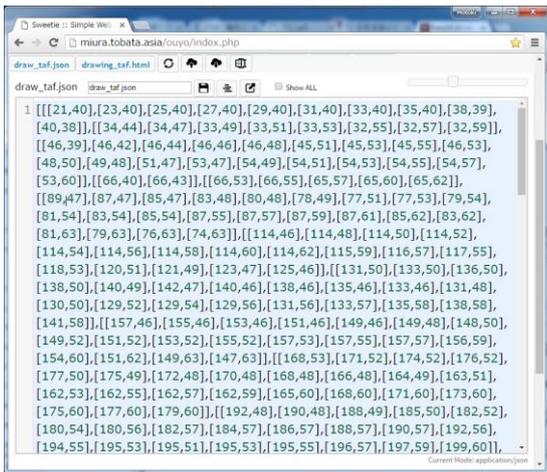


図 4 : JSON 形式による時系列筆記

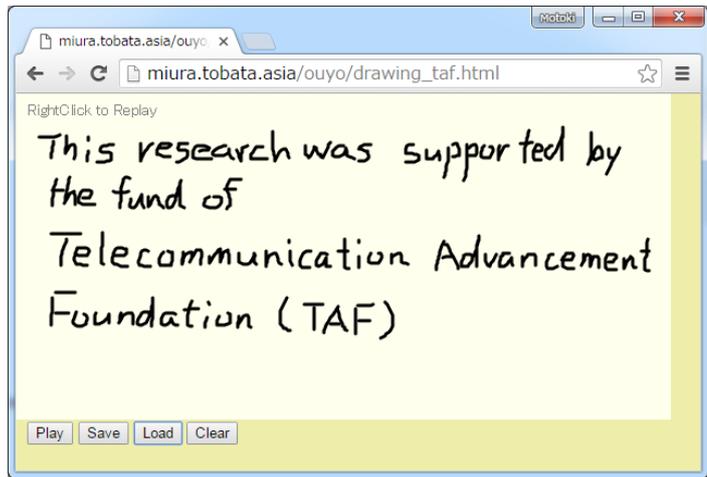


図 5 : 筆記閲覧・描画システムのインタフェース

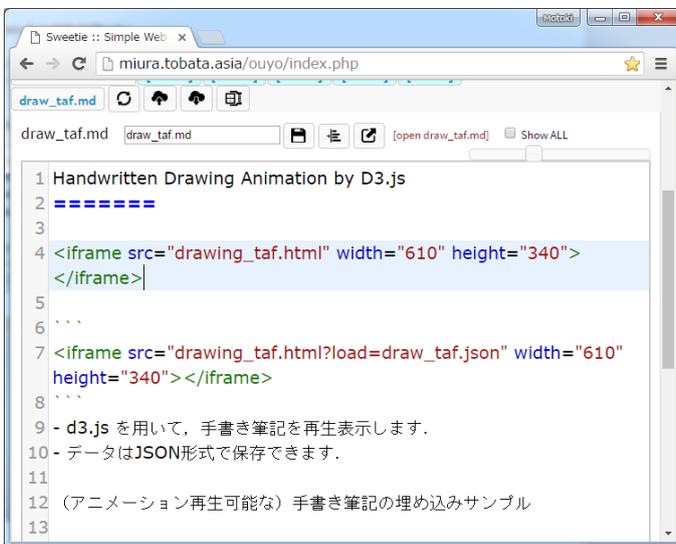


図 6 : Web ページへの埋め込み設定

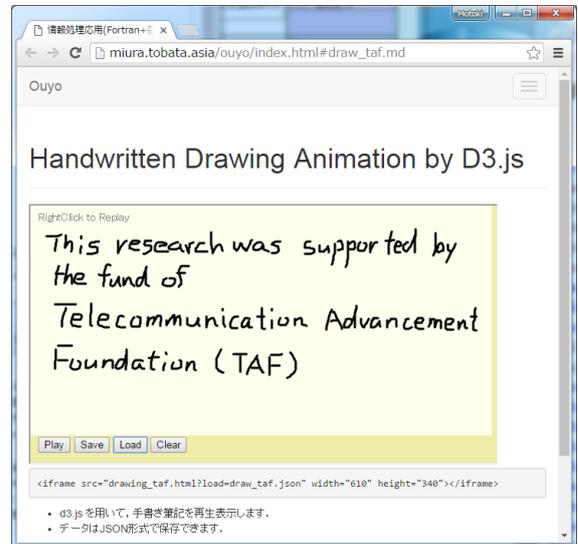


図 7 : 埋め込まれた Web ページの表示

いる。このような JSON 形式はシンプルな構造であるため、Meteor でも簡単に出力することが可能である。JSON 形式の筆記データファイルを、付録で示した Web ページによって読み込ませると、図 5 に示す表示を得ることができる。

図 5 の Web ページにおいて、Play ボタンを押すと、JSON ファイルの筆記が 1 ストロークずつ、順番になめらかに再生される。(再生の様子は上記のサイトまたは動画：<https://youtu.be/g4bKpVdScgA> で確認できる)。また、図 5 の Web ページで、直接マウス等を用いて筆記を追加したり、ファイルに保存することもできる(注: 筆記をファイルに保存するには、Web サイトにファイルを置くための認証情報が必要となる)。なお、筆記再生の仕組みはデータに基づいてドキュメントを操作するための Javascript ライブラリ D3.js を用いて構築しており、SVG と CSS を利用して筆記の描画を行っている。

このような筆記共有 Web ページやコンテンツについて、学生でも簡単かつ手軽に作成・編集・共有できることを目的として、Web ブラウザ上で動作するエディタを利用した Web 開発環境 Sweetie (Simple Web Editor Utilities) を構築した。Sweetie は、Web ブラウザ上で動作するエディタ (CodeMirror) とサーバサイドのスクリプトによって、Web サイトのコンテンツを Web ブラウザ上で直接編集・修正することができる。また、マークダウン記述で簡単に Web サイトを構築できる MDWiki[6] を利用しているため、HTML による記述と、マークダウン記述を混在させたページも作成可能である。図 6 に、マークダウン記述と HTML

記述を混在させた例を示す。エディタのシンタックスハイライト機能により、マークダウン記述や HTML 要素・属性が色付けされているため、初心者でも編集ミスに気づきやすい。図 7 は、図 6 のコンテンツを Web ブラウザで表示した例である。このように、図 5 の筆記閲覧インタフェースを埋め込んだ Web ページを簡単に構築できる。

Sweetie では、CTRL+S による保存や、カット・コピー・ペースト等のショートカットキーが利用できるため、通常のエディタと同様の操作感で Web ページの直接的な編集が行える。また、クリップボード画像の貼り付けや、画像ファイルのドラッグ&ドロップによる貼り付けを行うと、画像をサーバにアップロードしたうえで、画像をページ内に表示するためのマークダウン記法または HTML 要素を自動挿入する機能を備えている。これまでに、大学生を対象とする複数の講義やゼミ等で利用し、Web ページ構築や、Web アプリケーション構築演習に利用している。管理インタフェースを含む Sweetie のソースコードは GitHub で公開している。<https://github.com/miuramo/sweetie>

## 6 おわりに

Web 技術を活用した柔軟で効果的な遠隔講義向け筆記共有システムの研究として、デジタルペンと WebSocket 技術を用いた、遠隔地における紙への筆記をリアルタイムに Web ブラウザ上で共有・閲覧できるシステムの開発と、筆記をアニメーション再生可能な状態で Web アクセス可能にするための筆記閲覧・描画インタフェースの構築について述べた。また学生が筆記情報を含む Web コンテンツを簡単かつ手軽に作成・編集・共有できる Web 開発環境 Sweetie を構築した。今回の研究で構築したこれらの仕組みによって、手軽に思考を表出できる手書き筆記というメディアを、Web という強力な情報共有基盤に接近・接続することができるため、遠隔地間を含むコミュニケーションやコラボレーションの促進に寄与することが期待できる。

### 【参考文献】

- [1] Galusha, J. M.: Barriers to Learning in Distance Education, Interpersonal computing and technology: An electronic journal for the 21st century, Vol. 5, No. 3/4, pp. 6–14 (1997).
- [2] 三浦 元喜. デジタルペンによる同期型遠隔講義の活性化システム. 情報処理学会インタラクション 2014, pp. 657–660, 2014.
- [3] 三浦元喜, 杉原太郎, 國藤進: 一般教室での日常的利用を考慮したデジタルペン授業システムの改良, 日本教育工学会論文誌, Vol. 34, No. 3, pp. 279- 287 (2010).
- [4] 杉原太郎, 三浦元喜: 高校の数学授業実践を通じたデジタルペンシステムの効果, 情報処理学会論文誌, Vol. 54, No. 1, pp. 192–201 (2013).
- [5] Meteor: Build Apps with Javascript, <https://www.meteor.com/>. (2016 年 6 月 29 日確認)
- [6] Timo Drr. MDWiki – Markdown based wiki done 100% on the client via javascript. <http://mdwiki.info/>. (2015 年 12 月 17 日確認)

〈 発 表 資 料 〉

題 名	掲載誌・学会名等	発表年月
Sweetie: 学生計算機上で動作する協働作業に適した軽量 Web エディタ環境	情報処理学会情報教育シンポジウム(SSS2015), 117-123	2015. 8
Web 技術を活用した柔軟でオープンなデジタルペン筆記共有基盤	第 23 回インタラクティブシステムとソフトウェアに関するワークショップ WISS 2015, 147-148	2015. 12
お手軽 Web 開発環境 Sweetie	情報処理学会インタラクシオン 2016, pp. 281-284	2016. 3
Sweetie: Lightweight Web Authoring Environment	20th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems	2016. 9
Web 技術を活用したインタラクティブな情報教育環境の構築と実践	電子情報通信学会 ET 研究会, Vol. IEICE-116, No. 85, 19-24	2016.6
Browsing Methods for Multiple Online Handwritten Note Animations	The Eighth International Conference on Collaboration Technologies (CollabTech2016)	2016.9

付録：筆記閲覧・描画インタフェースのソースコード

参考サイト：<http://bl.ocks.org/mbostock/f705fc55e6f26df29354> <http://bl.ocks.org/mbostock/4342190>  
<http://bl.ocks.org/duopixel/4063326>

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script src="jq/external/jquery/jquery.js"></script>
    <style>
      .line {
        fill: none;
        stroke: #000;
        stroke-width: 3px;
        stroke-linejoin: round;
        stroke-linecap: round;
      }
      body {
        margin: 0px;
        background: #eea;
      }
    </style>
  </head>
  <body>
    <svg width="600" height="300">
      <rect style="fill:#ffe;margin:0px;" width="100%" height="100%"></rect>
      <text x="10" y="20" font-family="sans-serif" font-size="14px" fill="gray">RightClick to Replay</text>
    </svg><br>
    <input name="playB" type="button" value="Play" onclick="d3.selectAll('path').each(anim);" />
    <input name="saveB" type="button" value="Save" onclick="save();" />
    <input name="loadB" type="button" value="Load" onclick="load();" />
    <input name="clearB" type="button" value="Clear" onclick="sclear();" />
  </body>

```

```

<script src="//d3js.org/d3.v3.min.js"></script>
<script>
  var strokes = [];
  var activeStroke;
  var activeLine;
  var prex = 0, prey = 0;
  var filename = "draw.json";
  var renderPath = d3.svg.line()
    .x(function(d) { return d[0]; })
    .y(function(d) { return d[1]; });

  var svg = d3.select("svg")
    .call(d3.behavior.drag()
      .on("dragstart", dragstarted)
      .on("drag", dragged)
      .on("dragend", dragended));

  function dragstarted() {
    var mary = d3.mouse(this).map(Math.floor);
    activeLine = svg.append("path").datum([]).attr("class", "line");
    activeLine.datum().push(mary);
    activeStroke = [];
    activeLine.attr("d", renderPath);
    prex = mary[0]; prey = mary[1];
  }

  function dragged() {
    var m = d3.mouse(this);
    var mary = m.map(Math.floor);
    if (Math.abs(prex-mary[0])>1 || Math.abs(prex-mary[1])>1 ){
      activeLine.datum().push(mary);
      activeLine.attr("d", renderPath);
      activeStroke.push(mary);
      prex = mary[0]; prey = mary[1];
    }
  }

  function totalLength(v){
    return v.node().getTotalLength();
  }

  function dragended() {
    activeLine = null;
    strokes.push(activeStroke);
  }

  function anim(d, i){
    var v = d3.select(this);
    v.attr("stroke-dasharray", (totalLength(v)) + "," + (totalLength(v)+1))
      .attr("stroke-dashoffset", (totalLength(v)+1))
      .transition()
      .duration(500).delay(i*500)
      .ease("linear")
      .attr("stroke-dashoffset", 0 );
  }

  function save(){

```

```

$.ajax({
  url: "__save.php",
  type: "post",
  data: {file: filename, code: "" },
  timeout: 3000,
  success: function(result, textStatus, xhr){
    reallysave();
  }
});
}
function reallysave(){
$.ajax({
  url: "__save.php",
  type: "post",
  data: {file: filename, code: JSON.stringify(strokes) },
  timeout: 3000,
  success: function(result, textStatus, xhr){
    console.log("save success : "+strokes.length+" strokes.");
  }
});
}
function load(){
$.getJSON(filename, function(json){
  json.forEach( function(ary){
    strokes.push(ary);
    svg.append("path").datum(ary).attr("class", "line")
      .attr("d", renderPath);
  });
});
}
function sclear(){
  strokes = [];
  d3.selectAll("path").remove();
}
svg.on("contextmenu", function(){
  d3.event.preventDefault();
  d3.selectAll('path').each(anim);
});

$(document).ready(function(){
  $.ajaxSetup({cache: false});
});

</script>
</body>
</html>

```