

# 賢い電子黒板の作り方

田中 二郎<sup>1,a)</sup>

**概要：**電子黒板のソフトウェアは、メニューを選んで作業をおこなうのがほとんどである。しかし、メニューや各種アイコンは生徒の注意をそらし、授業に集中できなくしていると思われる。そこで、メニューやアイコンを一切使用しない電子黒板ソフトウェアを作成した。さらに、電子黒板の表示を生徒の端末に配信する仕組みも用意した。また、そこで使われるデータを追加・編集することにより、独習できる教材とすることができる枠組みを準備した。

**キーワード：**プログラミング・シンポジウム，夏，予稿集，電子黒板

## 1. はじめに

教育の現場で電子黒板が使われることが多くなってきているが、ハードはともかく、付属のソフトは使い勝手が悪く、教育効果がおもったように出していない。特に、使い方が直感的でない、メニューの数が多すぎる、アイコンの意味がわからない等の声がある。また、対応している機種やOSも限られ、ソフトウェアの更新がされないなどの不具合も散見される。

そこで、従来のホワイトボード程度の難易さで使える、高機能の電子黒板ソフトを作成した。本ソフトでは、メニューを排し、音声入力やマウス操作のみを使用している。また、教師画面を生徒画面に配信する機能も用意した。さらに、配信用データを編集することにより、アニメーションなどの機能も実現可能である。

## 2. よくある電子黒板のソフトウェア

市販の電子黒板のソフトウェアは、ほとんどメニューを呼びだして描画などの動作をおこなうも

のである。線分をひくだけでも、メニューを呼び出し、線分を選び、太さや色を選んで描く必要がある。(図1)

また、円の場合は、対角を指定して円を描くのが一般的である。これは、板書のときのような書き方ではなく、思ったとおりの円を描くのが難しい。(図2)

手書き文字入力となると、ひどいものではOSの手書き文字入力ツールを呼び出している。(図3)



図1 描画メニューの例

<sup>1</sup> 開智国際大学

<sup>a)</sup> jiro@kaichi.ac.jp

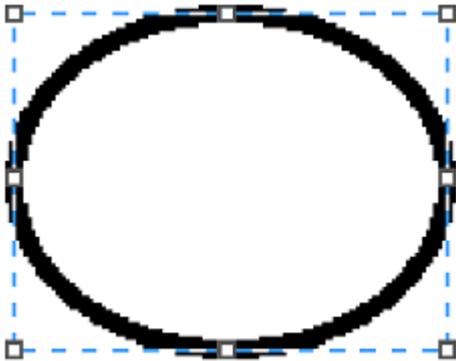


図 2 対角で円を描く



図 3 手書き文字認識パレット

### 3. 提案するUI

電子黒板で可能なアクションは、クリック、ダブルクリック程度である。

そこで、ダブルクリック後、音声入力を有効とし、認識した文字列を、ダブルクリックした点を左上として表示する。文字の大きさは板書に適したデフォルトの大きさである。(図4)

また、ダブルクリック後にマウスで描いたものは手書き文字認識エンジンに送り、認識した文字列を表示する。この場合、表示場所は手書きされた領域とし、文字の大きさは描かれた文字にあわせるようにする。(図5)

普通にマウスボタンを押しながら描くと、図形認識エンジンにより、線分もしくは円として描画する。(図6)

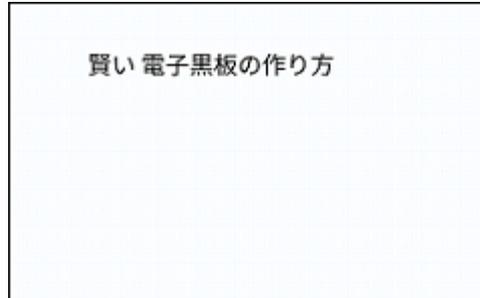


図 4 音声認識による文字入力

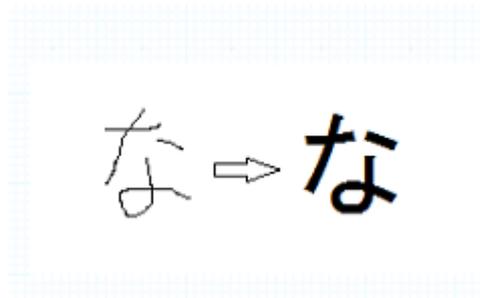


図 5 手書き文字入力（認識前と認識後）

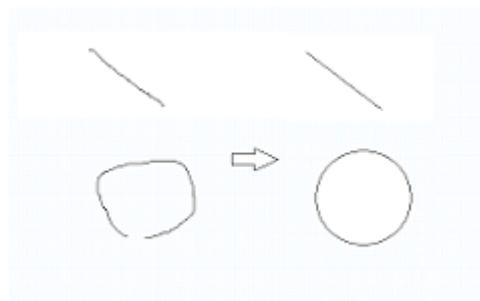


図 6 線分と円の描画（認識前と認識後）

このとき、始点や終点、円の中心や半径などのパラメータは、あらかじめ指定しているグリッドにあわせるか、既存の線分上、もしくはその中点、垂線となる点、円周上の点、円に接するようになる点など、過去に描いた要素と関係があるようなパラメータを選ぶようにすることで、まとまった作図とな

るようにすることができる。

描いた文字列や図形は、ドラッグすることにより移動できる。逆を言えば、既存の要素上でマウスボタンを押しても、線分や円の描画にはならない。消去は、画面外に移動することにより実現している。

いずれの場合も、メニューやアイコン、選択範囲などの表示はしない。そのため、生徒の視線がずれることなく、注意が散漫になることもない。

#### 4. タッチスクリーン入力

マウスを使つての文字入力や図形描画など以外に、タッチスクリーンによる描画もサポートしている。マウスによる操作と違い、フリーハンドの描画を赤色で表示している。描画した線は、最後に描画してから5秒後に、自動的に消去される。(図7)

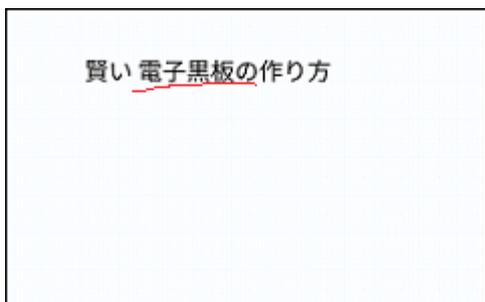


図7 タッチスクリーンによる描画(赤色)

#### 5. 生徒端末への配信

教師端末からのデータはサーバに送られ、それがスクリプトとしてサーバに記憶され、それを生徒端末が非同期に読みだすことにより、教師画面と同じもの(タッチスクリーンによる描画を除く)が、生徒端末に表示される。

通信および表示はブラウザ上のJavaScriptによりおこない、特別なソフトウェアは必要ない。サーバはデータ管理と送出力のみの簡易なCGIスクリプトだけで、この機構を実現している。(図8)

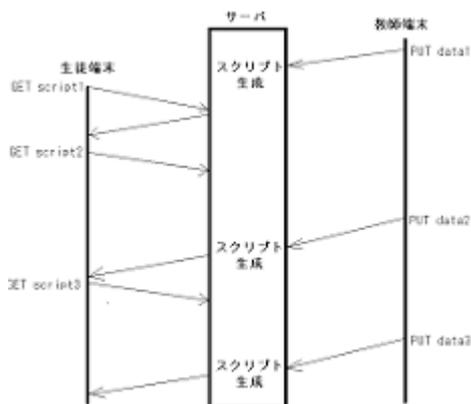


図8 生徒端末、教師端末とサーバとのやりとり

#### 6. スクリプト

教師端末の操作によって生成されるスクリプトは、教師端末の操作に対応している。

音声や手書き文字認識による文字入力は、対応するHTML要素を挿入する命令となる。図形入力は、キャンバスに対する描画命令となる。キャンバスが存在しないときは、キャンバス要素を挿入する命令の後に、描画命令を発行する。要素の移動は、移動命令を発行する。また、教師端末の操作によって生成される命令以外にも、種々の命令を用意している。

命令の最初の1文字が識別子で、命令の種類をあらわす。

- + HTMLの挿入。以後に書かれているものを現在のbodyに追加する。
- 要素の表示/非表示。指定されたidの要素が表示されていれば非表示(styleのdisplayをnone)にする。非表示であれば表示(styleのdisplayをblock)にする。
- ! 任意のJavaScriptの実行。
- = 要素のパラメータの変更。指定されたidの要素を変更する。
- \* 要素の移動。style.top, style.left, width, heightを変更する。

- @ 描画
- \$ アニメーション
- % 音声再生. 再生するファイルを指定する.
- : 時間待ち. mS 単位で待ち時間を指定する.

## 7. 描画コマンド

以下の描画コマンドをサポートしている.

- clear** キャンバス内消去.
- color** 色指定. 色相 (0-360°) もしくは RGB(0-255) で指定する.
- width** 線分の幅を指定する.
- origin** 極座標の原点を指定する.
- originr** 極座標の原点を, 現在の極座標を使って指定する.
- line** 線分を引く. 連続して指定すると, 折れ線を引く.
- liner** 極座標で点を指定し, 線分を引く.
- fill** 多角形を塗りつぶす.
- fillr** 極座標で点を指定し, 多角形を塗りつぶす.
- rect** 始点と大きさを指定して, 長方形を描く.
- rectfill** 始点と大きさを指定して, 長方形を塗りつぶす.
- arc** 円弧を描く.
- arcr** 中心を極座標で指定し, 円弧を描く.
- arcfill** 円弧を扇型に塗りつぶす.
- arcfillr** 中心を極座標で指定し, 円弧を扇型に塗りつぶす.
- curve** 4点を指定し, ベジエ曲線を描く.
- curvefill** 4点を指定し, ベジエ曲線内を塗りつぶす.
- font** フォント名を指定する.
- size** 文字の大きさを指定する.
- text** 指定した場所に, 文字を描画する.
- N** タートルグラフィックス: 北 (上) を向く.
- H** タートルグラフィックス: 指定した点の方向に向く.
- RT** 右に回転する.
- FD** (ペンを上げて) 移動する. 続けて指定すると, (ペンを下げて) 線分を描く.

## 8. アニメーション

描画コマンドのうち, 以下のものはアニメーションさせることができる.

**color, width, origin, line, liner, fill, fillr, rect, rectfill, arc, arcr, arcfill, arcfillr, curve, curvefill, text, N, H, RT, FD**

アニメーションでは, 各パラメータの始値と終値を指定し, その間を線形補完しながら描画している. これにより, なめらかに, 位置や大きさ, 色を変化させるアニメーションを表示できる.

コマンドの中では, 「H」など, 一見アニメーションに無関係なものもあるが, 図形全体を描画する都合上, 必要なものである.

## 9. 応用

このUIに, 人工知能を利用したバックエンドを接続することで, 筆算の計算から数式の変換といった高度なものまで実現できると面白い. 特に, 「この」と発話しながら点を指し示すことで, その点を理解させるなど, 音声認識との相互作用を組み込む予定である. これが実現すると, 「円に内接する三角形のこの点が直角なら, 対する辺は円の中心をとる」という発話から, 適切な図形を描画しアニメーションすることが可能となろう.

## 10. まとめ

メニューを廃したUIにより, これまでの (電子ではない) 黒板のように使用でき, 電子黒板を使用するハードルがさがった. さらに, 音声入力や手書き文字認識により文字を清書でき, 線分や円もきれいに描ける. また, スクリプトを再利用したり編集することにより, 教材として利用できるようになった.

#### 質疑・応答

Q (伊地知) 未踏で電子黒板で数式や図形処理をしてした。

A 色々なものを, このシステムに統合していきたい。

<https://www.ipa.go.jp/files/000005999.pdf> ?

Q (久野) 先生に独特なスキルを要求するのでは?

A 従来のホワイトボードに加えて, クリック・ダブルクリックの操作のみを加えただけ。アニメーションなど, 高度なことは, できる人だけやればよい。

Q (久野) パワーポイントでよいのでは?

A 教員がパフォーマンスすることによって, 生徒の理解をすすめる。

C (伊地知) アニメーション機能は, 従来のスライドではできないことである。

A 点をつかんで移動させるなどの機能も追加する予定。

Q それを生徒ができるようにすればよいのでは?

A 別に教員側に限らない。生徒側も同様のソフトを使えばよい。

Q (山之上) 「P2P 技術を利用した分散システム上の実時間操作共有システム」というのがある

A 読ませていただきます。→情報処理学会論文誌, 46(2): 392-402

Q データを集積していけば?

A 教科書会社がやれば, すごいと思う

Q (久野) 先生がやらなければいけないことのイメージがわからない

A ダブルクリックして発話すると, 音声認識して画面に表示される。マウスで線や円を描くと, 清書した図形を描画する。

Q 色々できることがあると, 先生が理解できなくなるのではないか?

A 従来のホワイトボードでやっているとおり, 文字を書いたり, 線分や円を描くだけ。

Q 本当にシンプル?

A 内部コードは多機能だが, それを全部知って使

う必要はない。

Q (伊地知) 全部消去について, 巻物という概念を導入できないか?

A 将来の課題。

Q 音声はテキストになるだけ? コマンドとして使えない?

A 使えるがエスケープ問題がある。声の大きさとか, 別途キーによる指定とかが必要。