

視覚情報保障機能を備えた Web プラットフォームの開発

筑波技術大学 保健科学部 情報システム学科¹⁾ 保健学科鍼灸学専攻²⁾

伊藤裕二¹⁾ 成島朋美²⁾ 三宅輝久¹⁾ 大越教夫²⁾ 小野 東¹⁾

要旨:近年、教育現場の ICT 化に伴い、グループウェアなどの Web アプリケーションが導入されているが、視覚障害者がユーザーとなる場合、見え方の多様性に対応する必要がある。現在は、弱視者対応の画面表示機能を有する既製アプリケーションに既製スクリーンリーダーによる読み上げ機能を組み合わせて対応するのが主流であるが、読み上げを詳細に制御できないためシステムとしては不完全である。そこで、我々はサーバー・クライアント協調設計による音声読み上げと画面表示カスタム化による情報保障機能を備えた Web プラットフォームの開発を試みた。このプラットフォームを用いることで、視覚障害者向けオンライン教材配信システム、テストシステムなどの開発が容易になると考える。

キーワード:協調設計, Web プラットフォーム, 音声読み上げ, 画面表示カスタム機能

1. まえがき

近年、教育現場の ICT 化に伴い、グループウェアや Moodle [1] などの学習管理システム (Learning Management System : LMS) に代表される Web アプリケーションが導入されている [2]。視覚に障害のある学生が鍼灸、理学療法、情報システムを専攻する筑波技術大学保健科学部でも効率的な情報伝達の実現を目指して様々な試みがなされている。視覚障害者がユーザーとなる場合、このような Web アプリケーション構築にあたって、視覚障害者の見え方の多様性に対応するユーザーインターフェース (UI) を具備することが重要な要件となる。つまり、音声や画面表示機能を駆使してユーザー個人毎に適した情報保障機能を備える必要がある。現在は、弱視対応の画面表示機能を有する既製、または、独自開発のアプリケーションに既製スクリーンリーダー (PC 用画面読み上げソフトウェア) による読み上げ機能を組み合わせて対応するのが主流である。しかし、この構成では、必要でない情報を毎回読み上げる反面、重要な情報を読まないケースがあり [3]、システムとしては不完全である。クライアント側でデータを再構成して既製スクリーンリーダーとの親和性を高める試みがあるが、一部機能が動作しなくなるという問題点が報告されている [3]。

そこで、我々はサーバー・クライアント協調設計による読み上げ機能と画面表示カスタム機能を有する Web プラットフォームの開発を試みた。このプラットフォームを用いることで、視覚障害者向け教材配信、オンラインテスト・アンケートシステム、情報共有サイトなどの開発が容易になると考える。

2. 要求仕様

まず、大きな方針として、1) 視力障害の程度に関係なく操作可能なユーザーインターフェース (UI) を備えること、2) 可能な限りオープンソース・ソフトウェアやフリーソフトウェアを活用してコストを抑えること、の2項目を掲げた。1) に関しては、全盲者に対する音声案内機能、弱視者に対する文字サイズ、背景色の調整機能、及び、入力箇所の案内などの入力支援機能を要求性能とした。音声案内機能に関しては、従来市販のスクリーンリーダーにより対応することが多かった。しかし、スクリーンリーダーでは必要のない情報まで読み上げてしまう反面、本当に必要な情報を読み飛ばしてしまうこともあるため、システム独自の音声案内機能が必要であると考えた。一方、2) に関しては、開発自体のコストを抑える意味もあるが、将来の可能性として、盲学校等に頒布することになった場合の導入側の金銭的負担を軽減するためでもある。表 1 に要求仕様を示す。

表1 要求仕様一覧

機能	性能又は指標
全盲者対応 UI	音声案内
弱視者対応 UI	文字サイズ及び配色の調整 (ユーザー毎に設定可)
限定アクセス	ユーザー毎の権限に従った各種 データへのアクセス
データ管理	SQL により RDBMS サーバーを操作
グループウェア	グループによる協調作業を支援する ツール

情報保障関連の要求には、全盲者対応 UI と弱視者対応 UI がある。限定アクセスは一般的なシステムの要求にも含まれるが、担任教師、一般教師、担当学生（ある役割を与えられた学生）、一般学生という教育現場特有の役割に応じた権限を設定する必要がある。また、オンラインテストなど多量のデータを扱うアプリケーションに対応するため汎用性の高い SQL 言語による RDBMS サーバーの操作を要求項目に入れた。更に、学部や学科内での情報共有システムとしても利用できるようにグループウェア機能も要件に含めた。

3. 開発

3.1 開発環境の選定及び整備

システム構成を図 1 に示す。PC 環境としては、教育現場で採用実績のある Windows PC（サーバー、クライアント側共に OS は Windows XP）を用いた。また、基礎プラットフォームとして FreeStyle Wiki (FS Wiki) [4] を用いることにした。FS Wiki は Wikipedia に代表される Wiki の派生である。Wiki は汎用 Web ブラウザを用いて操作可能であるため、ユーザー側では専用ソフトウェアをインストールする必要がない。また、FS Wiki はオープンソースであるため、必要に応じて機能モジュール（プラグインと呼ぶ）を追加することにより機能拡張が可能である。開発時、運用時共、プラットフォームを Windows XP、Apache 2.2 Web サーバー上で CGI (common gate interface) として動作させた。開発言語は FS Wiki にあわせて Perl を採用し、クライアント側の UI を強化するために JavaScript を用いた。また、データベースとしては、MySQL を利用した。そして、音声案内を実現するための音声合成エンジン（形態素解析を含む）としてマイクロソフト社が提供している MS Agent をクライアント側 PC にインストールした。

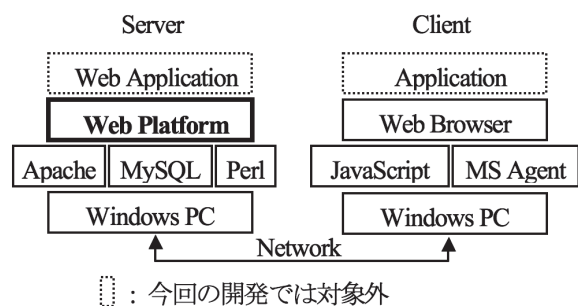


図 1 システム構成（概念図）

3.2 実装

3.2.1 Web ページのデザイン

視覚障害者が内容を理解し易いように一度に表示する情報量をひとつの意味のかたまりになるように抑える。例えば、データベースの情報を管理するページでは、初期画面では検索、既入力レコードへのリンク、新規レコード作成ボ

タンを表示する。そして、既入力レコードのリンクが選択された場合は、レコード詳細、レコード更新ボタン、レコード削除ボタン、初期画面へのリンクを表示する。また、新規レコード作成ボタンが押された場合は、レコード入力欄、初期画面へのリンク、レコード追加ボタン、初期画面へのリンクを表示する。更に、ページの構成を固定部分（ページに依存せず常に同じ情報が提示される部分）と可変部分（ページ依存で変化する部分）に分け、固定部分に操作案内へのリンク、他ページへのリンク等を配置することで一貫性のある操作が可能になると考える。電子カルテを想定して作成したページのデザイン例を図 2 に示す。ページ名は”患者情報”、”予約”から”ユーザー設定”までが他ページへのリンクを示す固定部分、それ以下が可変部分である。



図 2 ページのデザイン例（電子カルテを想定した場合）

3.2.2 全盲者対応 UI

全盲者対応の UI は、音声合成による案内と Web ページ内にアンカー（a タグ）に代表される誘導用タグを配置することにより実現される。

A) 音声合成エンジン

音声合成エンジン（形態素解析を含む）にはマイクロソフト社が提供している MS Agent を用いた。ページ読み込みの際の初期化ルーチンで MS Agent を起動させる。Agent は同時に複数起動することができ、それぞれに声色を設定することができる。実際には、立体感のあるアニメキャラクターが画面上に現れて指定したテキストを吹き出し表示し、読み上げる（キャラクターを表示させない設定も可能）。読み上げ速度、効果音の有無はクライアント PC 側で調整可能である。

B) 現入力値の表現

基本的には、クライアント PC の状態を監視するために常用される JavaScript の setTimeout 関数を利用する。状態判定と処理を行う関数に setTimeout 関数を用いて再帰呼出しすることで状態判定と処理が次の初期化時（例えばページ更新）まで有効となる。テキストボックスから入

力した値を音声出力する擬似スクリプトを図3に示す。現在フォーカス中の部品を activeElement で取得し、その属性値である type で処理を切り替え（例えば、図3ではテキストボックスの処理を示す）、部品名 id と値 value を音声出力する。speakText (str) は文字列 str を読み上げる関数、speechNavi (aE) はフォーカス中の部品 aE 内において文字が追加、削除されたこと、仮名漢字変換されたことなどを判定し、それらを音声で通知する関数である。

```

var curr_aE=""; // 直近の activeElement
function loopTimer(){
  if(window.document.activeElement != undefined){
    var aE = window.document.activeElement;
    if(aE.id != undefined){
      if(curr_aE.id != aE.id){
        if(aE.type == "text"){
          speakText(aE.id+"の値は"+aE.value);
        }
      }
      if(aE.type == "text"){
        speechNavi(aE);
      }
    }
    curr_aE = aE;
  }
  timer = setTimeout("loopTimer()",30);
}

```

図3 入力テキストを音声出力する擬似スクリプト
(テキストボックスの処理例を示す)

ラジオボタン、チェックボックス、選択ボックスに関しても同様の処理を行う。テキストボックスに関しては、詳細な音声案内をするためにはカーソル位置を知る必要があるので JavaScript の createRange 関数及び moveStart 関数を用いて特定する。また、複数行入力可能な textarea に関しては、改行コードの扱いが煩雑になるためカーソル位置特定には更なる工夫を要した。最後に、編集時の値を読み上げる場合は、句読点など記号も含めて一字一句を正確に表現する必要があるため、"。" を"まる"などに置換する処理を読み上げ直前に施した。

C) 既入力値の表現

既入力値を読み上げるとき、特に問題となるのがカラム数が多いテーブルで、どのカラムを読み上げているかが分からなくなることである。また、市販のスクリーンリーダーではカラム数に関係なくデータを読み上げない場合がある。対処法として、Tab キーで必ず辿ることができる a タグを用いたリンクを各レコード（通常先頭カラム）に配置し、その a タグに特殊な type 属性 (link とした) を付けて識別できるようにし、name 属性で読み上げる情報を指定する。例えば、図2の患者 ID00001 の先頭カラムは以下のようにになっている。

```

<a type="link" name="患者 ID00001 伊藤裕二 男性"
href="id=1">00001</a>

```

そして、クライアント側では前述の JavaScript の loopTimer 関数内に type == link の場合に name 属性で指定された値を音声出力する処理を入れることで当該リンクがフォーカスされたとき音声案内が流れるようになる。このようにサーバー・クライアント間で独自にルールを決めて処理ができるという協調設計の利点を活かすことによって所望の音声案内が可能となる。

D) 漢字変換に伴う処理

全盲者が漢字を使う場合、変換候補の漢字を特定するための情報（主に訓読み）を音声案内する。そのために、まず、常用漢字程度（3000字）をカバーする辞書を作成した (yomi['藤']= "ふじ" の形式)。そして、前述の speechNavi 関数に漢字変換中であるか否かを判定する処理を実装し、漢字変換中のときに限り変換候補の漢字を一文字ずつ辞書中の読みに変換して音声出力する。

今後は、アプリケーション毎に適した辞書を整備する必要がある。例えば、電子カルテの場合は医学用語などの専門用語をカバーした辞書を使うことになる。辞書を作成する労力は要するが、確実に指定した読み情報を提供できる。

E) 誘導用タグの配置

音声案内を有効利用するため誘導用タグを配置して音声案内の箇所を Tab キー及び数個のアクセスキーのみで辿ることができるようにする。まず、可変部分の先頭にアンカー（先頭アンカー）を置き、そこに当該ページ名と内容に関する簡単な音声案内を入れる。具体的には a タグに特殊な type 属性 (navi とした) を付けて識別できるようにし、name 属性で案内情報を指定する。例えば、図2の可変部分の先頭（検索文字入力ボックスの前）には表示されない以下のようなタグが挿入されている。

```

<a type="navi" id="p" name="患者情報..."
href="top"></a><a name="#top">

```

そして、JavaScript の onload 時の処理に上記アンカーへの移動を指定することでページが変わる毎に音声案内が流れるようになる。また、可変部分の末尾にも同様のアンカーを配置し、これら2つのアンカーにはアクセスキー（例えば、Alt+U と Alt+S）を振る。これにより、可変部分の両端にアクセスキーで移動でき、その中間は Tab キー又は Shift+Tab キーにより内容に則した音声案内を辿ることができる。また、先頭アンカーから Shift+Tab キーで、別ページへのリンクを辿ることができる。

F) 誤入力防止

誤入力を防止するための工夫を施した。例えば、英数

字制約があるユーザー ID やパスワードを入力する際、誤って全角文字が入力されたとき、該当する英数字が存在すれば変換し、存在しなければ誤入力であることを音声で知らせる。これは、JavaScript の onkeyup イベント発生時に入力文字をチェックすることで実現できる。また、片仮名制約のある所に平仮名を入力してしまった場合も（その逆の場合も）同様に対処することができる。

3.2.3 弱視者対応 UI

今回の開発では、ユーザー毎に好みのスタイルシートを設定するしくみを用いてブラウザ独自の表示調整を実現した [5]。FS Wiki 自体は表示スタイルを記述する CSS (cascading style sheet) をサポートしているが、サイト内で設定できる CSS は固定されている。そこで、ユーザー情報に CSS を設定できるようにプログラムを変更することにより、個人毎に好みの配色や文字サイズで Web コンテンツを表示できるようにした。そして、弱視者の多様な見え方に対応すべく、数種の代表的な配色（例えば、背景：黒色、文字：白色）を定義した CSS ファイルを選択肢として準備した。ページ中の各要素（タイトル、ヘッダなど）の文字サイズは主たる要素である body 要素に対する相対値で表現する。背景：黒色、文字：白色の場合のスタイルシートの例を図 4 に示す。この例では、h1 で示されるページタイトルの文字サイズが body 要素に対して 180% に設定されていることがわかる。

```
body {
  color                :#FFFFFF;
  background-color    :#000000;
  font-family         : "MS ゴシック", sans-serif;
}
h1 {
  font-size : 180%;
}
h2 {
  font-size : 150%;
}
h3 {
  font-size : 135%;
}
h4 {
  font-size : 120%;
}
```

図 4 スタイルシートの例（背景：黒色、文字：白色）

文字サイズは配色と独立に設定できるように工夫し、body 要素 = 1em（文字 M の高さ）を 100% として 60% から 300% までの選択肢を設けた。body 要素の大きさはスタイルシートに記述されず、Wiki から動的に設定される。選定した配色と文字サイズはユーザー情報として記録するので、毎回同じ環境で閲覧することが可能である。この UI を用いて、背景：黒色、文字：白色、文字サイズ：170% を指定した場合のブラウザ表示を図 5 に示す（比較対象

として図 2 は背景：灰色、文字：黒色、文字サイズ：100%）。



図 5 背景：黒色、文字：白色、文字サイズ：170%の表示例

Web で利用される技術の標準化をすすめる団体 W3C (World Wide Web Consortium) は、Web コンテンツへのアクセシビリティ向上を目的として CSS に関する勧告 [6] を定めている。今回開発した弱視者対応 UI が同勧告に準拠していることを確認した。

3.2.4 限定アクセス

FS Wiki では、ユーザー登録ができ、また、ページ単位でビジター（未ログインのユーザー又は訪問者）とログインユーザーの閲覧権限、変更権限（作成、削除の権限を含む）が設定できる。しかし、これだけでは所望の限定アクセスを実現できないので仕様を拡張した。具体的には、ビジター (V) とログインユーザー (Y) 以外に、特権ユーザー群 (P、特別に指定されたログインユーザー) とページオーナー (O、ページ作成者) というオーナーシップ階層を設け、ページ毎に閲覧、変更の可否が設定できるようにした。一般的なページでは、ビジターには閲覧権限のみを与え、その他の階層（特にページオーナーに関しては常に）に対しては閲覧及び変更の権限を与える。

また、特定の属性をもったユーザーをグループとして定義できる機能を追加した。これと特権ユーザー群を併用するとアクセス管理が効果的になる。例えば、経理関係のページが複数あり、特定の経理担当者にだけ編集させたい場合、特権ユーザー群に経理担当者を指定して変更権限を与える。経理担当者が変わった時は、ページ毎に権限を変更する必要はなく、経理グループの構成員を変更するだけで完了する。

以上の方法によりページ単位での限定アクセスは実現できる。更に小さな単位（例えば、データベース中の 1レコード）のアクセス制御が必要な場合は該当するプラグインのスク립トを書き直して対応することになる。

3.2.5 データ管理

データ管理はRDBMSとしてのMySQLを利用することにより開発の負担を軽減した。PerlからMySQLサーバーへのインターフェースはDBIモジュールを用いて実装し、データ操作はSQLの発行により行っている。データのバックアップはmysqldumpを用いたバッチ処理をcron擬きのフリーソフトウェアにより自動で定期的に行うことで対処する。

データの真正性が求められる電子カルテなどのアプリケーションでは更新履歴を残すことが重要である。これには当初、Insert、Update、Delete時に自動的にログを残すTrigger機能を応用する予定であった。しかし、変更者(SQLを発行したユーザー)の情報をログに含める手段が見当たらなかったため、変更者を含めた更新履歴を残すPerlモジュールを開発した。

3.2.6 グループウェア機能

FS Wikiは本来グループウェアとして設計されているので、スケジュール管理、掲示板などの有用な機能がプラグインとして提供されている。今回は様々なアプリケーションへの対応に備え、2つのプラグインモジュールを独自に開発した。

1つ目は、予め宛先メールアドレス、送信時間、内容を設定し、電子メールで通知するreminderプラグインである。送信内容にはテキストファイルも指定できるようにし、その情報はWiki側でデータベース管理する。例えば、設定時間を1分単位とした場合、データベースにアクセスして現在時刻に近い(現在時±30秒の間)送信時間が設定されているレコードを抽出して電子メール送信する独立したPerlスクリプトをcron擬きのソフトウェアにより毎分実行する。時刻補正のためサーバーPCの内蔵時計を標準時刻に合わせるソフトウェア(フリーソフトウェアが利用できる)を定期的に自動で起動させることが望ましい。

2つ目は、オンライン承認を行うapproveプラグインである。承認者をユーザーIDで指定し、承認順通りに並べて承認者リストを作成する。承認案件をWikiのページとして作成し、それを指定してプロセスを開始すると順番通りに承認依頼の電子メールが送付される。メールで指示されたWikiページにある選択ボックスで承認・否認を行う。発議者及びCC(カーボンコピー)に指定されたユーザーには承認・否認される毎に通知メールが届いて進捗を確認できる。

4. 制約・問題点及び今後の対応

今回開発したWebプラットフォームはWindows PC(サーバー、クライアント側共にOSはWindows XP以降、ネットワークカード付属)とネットワーク環境さえあれば移植可能である。しかし、現時点では開発したJavaScript関数の一部に互換性の問題があるためWebブラウザはIE

(Internet Explorer)に限定される。しかし、学内等での運用では使用するクライアントPCを自前で設定できるので問題ないとする。将来的には、他のブラウザにも対応する予定である。

MS Agentの音質についてはやや抑揚に欠けるため、単語によっては聞き辛いことがある。この点に関しては、市販のスクリーンリーダーに僅かながら劣るが、調査した数種のフリーソフトウェアの中では最上位を争った。アプリケーションによっては、より高音質な音声合成エンジンに切り替える必要がある。また、形態素解析の性能に関しては、MS Agentは市販のスクリーンリーダーと同等以上であると評価した。

また、サーバー・クライアント協調設計による読み上げでは、詳細な設定ができるため声色の切替えなども可能になる。例えば、「」内のテキストを本文とは別の声色で読み上げるように設定すると電子図書の朗読などにおいてリアリティを増す効果が期待できる。

5. まとめ

視覚に障害をもつ学生が使うWebアプリケーションを構築するためのプラットフォームを開発した。今後は、このプラットフォームを用いて教材配信システムやオンラインテスト・アンケートシステムの開発を進める予定である。

謝辞

本研究は平成22年度文部科学省特別経費「視覚に障害をもつ医療系学生のための教育高度化改善事業」の支援を受けて実施した。

参考文献

- [1] <http://moodle.org/>
- [2] 例えば、三重大学 Moodle: <http://portal.mie-u.ac.jp/moodle10/>
- [3] 小林, 加藤, : グループウェア用音声ブラウザの開発とマルチモーダル誘導路システムの設置, 筑波技術大学テクレポ17(2), 66-71, 2010.
- [4] FreeStyle Wiki: <http://fswiki.sourceforge.jp/cgi-bin/wiki.cgi>
- [5] Y. Itoh, T. Urushihata, et al.: Development of Web Crawler and Database System for Visually Impaired Students - An Application to Career Support Web System -, NTUT Education of Disabilities 8: 13-20, 2010.
- [6] The World Wide Web Consortium (W3C): CSS Techniques for Web Content Accessibility Guidelines 1.0, 2000: <http://www.w3.org/TR/2000/NOTE-WCAG10-CSS-TECHS-20001106/>

Development of Web Application Platform for Visually Impaired Students

ITOH Yuji¹⁾, NARUSHIMA Tomomi²⁾, MIYAKE Teruhisa¹⁾, OHKOSHI Norio²⁾, ONO Tsukasa¹⁾

¹⁾Department of Computer Science

²⁾Course of Acupuncture and Moxibustion, Department of Health
Faculty of Health Sciences, Tsukuba University of Technology

Abstract: This report describes a Web platform that functions as the base system for Web applications intended for visually impaired students. As ICT (information and communication technology) gets pervasive in educational institutions today, the Web applications such as groupware have been widely introduced. Special care must be taken to provide information assurance when building applications for visually impaired people. Conventionally, Web applications usually comprise an off-the-shelf screen reader and an off-the-shelf application that supports display customization (e.g., screen magnification). However, conventional Web applications are not optimal in the sense that they do not provide an accurate text-reading functionality. Therefore, we propose and develop a Web platform that provides the following information assurance capabilities: text reading by using a server-client co-design scheme, and the display customization. It is anticipated that this Web platform would promote the development of new Web applications for visually impaired students such as online material distribution systems and e-Learning systems.

Keywords: server-client co-design scheme, Web platform, screen reader, display customization