

K-056

## 視覚障害学生用 Web アプリケーション構築のためのプラットフォームの開発 Development of Web Application Platform for Visually Impaired Students

伊藤裕二†  
Yuji Itoh

成島朋美†  
Tomomi Narushima

三宅輝久†  
Teruhisa Miyake

大越教夫†  
Norio Ohkoshi

小野 東†  
Tsukasa Ono

### 1. まえがき

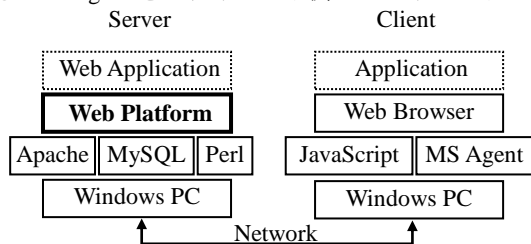
近年、教育現場の ICT 化に伴い、グループウェアや学習管理システム (Learning Management System : LMS) などの Web アプリケーションが導入されている。視覚障害者がユーザーとなる場合、このような Web アプリケーション構築にあたって、視覚障害者の見え方の多様性に対応するユーザーインターフェース (UI) を具備することが重要な要件となる。現在は、弱視者向けの画面表示調整機能を有する既成、または、独自開発のアプリケーションに既成スクリーンリーダー (PC 用画面読み上げソフトウェア) による読み上げ機能を組み合わせて対応するのが主流である。しかし、この構成では、画面情報を適切に読み上げないケースがあり [1]、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、視覚障害者向け情報共有アプリケーションなどの開発が容易になると考える。

### 2. 開発

#### 2.1 要求仕様、開発環境の選定及び整備

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) を備えることを要件とした。具体的には、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、入力箇所案内などの入力支援機能を要求性能とした。

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) [2] を用いることにした。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。開発時、運用時共、プラットフォームを 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。そして、音声案内を実現するための音声合成エンジン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。



⊔: 今回の開発では対象外

図 1 システム構成 (概念図)

### 2.2 実装

#### 2.2.1 全盲者対応 UI

全盲者対応 UI は、音声合成による案内と Web ページ内にアンカー (HTML の 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する。

#### A) Web ページのデザイン

全盲者が内容を理解し易いように一度に提示する情報量を抑え、かつ、関連する項目をまとめて配置する。例えば、データ情報管理においては、初期画面では検索、レコードへのリンク、新規作成ボタンを提示する。そして、レコードのリンクが選択された場合は、レコード詳細、更新ボタンなどを提示する。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。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。"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)

#### B) 現入力値の読み上げ

フォーカス中、かつ、編集時のテキストボックスから入力した値を音声出力するための JavaScript 擬似スクリプトを図 3 に示す。

```

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 編集時の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)

†国立大学法人 筑波技術大学 保健科学部

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)、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。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。

### C) 既入力値の読み上げ

編集途中でない既入力値を読み上げるとき、特に問題となるのはカラム数が多いテーブルで、読み上げているカラムが特定できないことである。対処法として、`Tab` キーで辿ることができる `a` タグを用いたリンクを各レコードの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り所望の音声案内が可能となる。

### D) 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を振る。これにより、可変部分の上下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` キーにより設計通りの音声案内を聞いて辿ることができる。

### 2.2.2 弱視者対応 UI

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 Wiki のコアプログラムを変更し、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 [3]。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。設定し

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。例として、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 4 に示す (比較対象の図 2 は背景:灰色、文字:黒色、文字サイズ:100%)。また、今回開発した弱視者対応 UI が、アクセシビリティ向上を目的とした CSS に関する勧告 [4] に準拠していることを確認した。

患者情報			
<a href="#">予約</a> <a href="#">診察記録</a> <a href="#">会計</a> <a href="#">ユーザー設定</a>			
			<a href="#">検索</a> <a href="#">リセット</a> <a href="#">新規</a>
6件すべて			
患者ID	氏名(漢字)	性別	生年月日(年齢)
00001	伊藤裕二	男性	2010年 5月 19日
00002	伊藤裕二	男性	2010年 5月 21日
00003	伊藤裕二	女性	2010年 5月 21日
00004	伊藤裕二	男性	2010年 5月 21日
00005	伊藤裕二	男性	2010年 5月 21日
00006	成島朋美	女性	2010年 5月 21日

図 4 背景:黒色、文字:白色、文字サイズ:170%の表示例

## 3. 制約・問題点

今回開発した Web プラットフォームは Windows PC とネットワーク環境さえあれば移植可能である。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はないと考える。

## 4. まとめ

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ことを特長とする Web アプリケーション構築の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視覚障害者向け情報共有アプリケーション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] 小林ほか, “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”, 筑波技術大学テクノレポート, Vol.17, No2, pp. 66-71, 2010.
- [2] FreeStyle Wiki: <http://fswiki.org/>
- [3] Y. Itoh et al., "Development of Web Crawler and Database System for Visually Impaired Students - An Application to Career Support Web System -", NTUT Education of Disabilities, Vol. 8, pp. 13-20, March 2010.
- [4] W3C: CSS Techniques for Web Content Accessibility Guidelines 1.0, 2000: <http://www.w3.org/TR/2000/NOTE-WCAG10-CSS-TECHS-20001106/>