

スクリーンリーダによる PDF 文書へのアクセシビリティについて

吉本 浩二^{*1} 渡辺 哲也^{*2}

A Survey on Accessibility to PDF Files by Screen Readers

Koji Yoshimoto^{*1} and Tetsuya Watanabe^{*2}

Abstract - For its convenience, Adobe PDF (Portable Document Format) files have come into wide use in disseminating public information. It has been, thus, warned that PDF files are raising another accessibility problem to screen reader users. With this background, we conducted a survey on accessibility to PDF files by screen readers. Accessibility to PDF files presents several aspects such as text sequence for re-use, keyboard navigation, combinational use with screen readers and how to make PDF files logically constructed. This paper describes these PDF accessibility issues and attempts to provide considerations about requirements of accessibility solution for electronic documents generally.

Keywords : People with visual impairments, Screen reader, Accessibility to PDF files, The Internet

1. はじめに

近年、インターネットによる情報通信は視覚障害者にとって不可欠なものとなってきている。従来は印刷物の形態で提供されてきた行政情報、新聞、雑誌など社会情報の多くが、インターネット上で電子情報としても提供されているので、スクリーンリーダを使って視覚障害者もこれらへ独力でアクセス可能となったためである。

インターネット黎明期は、クライアントブラウザの実装や通信速度等の問題により文字中心の Web ページが中心であった。しかしその発展に伴い、Web コンテンツは画像やレイアウトといった視覚的装飾を帯び、加えて単純な静的表示にとどまらず入力操作による変化や動画などマルチメディアの性質を帯び、視覚障害者にとって Web アクセシビリティの問題をはらむようになった。

Web アクセシビリティを確保するため W3C からガイドラインが示され^[1]、これを一般に伝える書籍・Web サイトも文献[2]をはじめとして多数公開されている。また Web 上の情報を効率的に伝える方法についても数多く研究されている^[3-5]。しかし、近年 Web 上で多数見られる PDF 文書のアクセシビリティに関する報告は、文献[6]のコラム (p.140-141) を除けば、国内の電子・情報系の学会・研究会や Web 上にもほとんど見あたらない。その理由として、PDF 文書のアクセシビリティは Adobe 社単独の問題と捉えられていることや、Acrobat のバージョン 5 以降はアクセス可能という認識が Web アクセシビリティの専門家の間にもあることが挙げられる^[7]。

筆者らが行った調査は、視覚障害者による行政や公的団体の Web ページの利用率が高いことが示した^[8]。一方、国内における電子政府・電子自治体の推進により^[9]、PDF 文書による情報提供の範囲はますます広がり、視覚障害

者がこれに遭遇する機会も増加している。既に官報^[10]や各種白書の付属 CD-ROM は PDF 文書で提供されており、これを視覚障害者が読めない問題が指摘されている^[11]。

そこで本稿では、既存のスクリーンリーダで PDF 文書にどこまでアクセス可能か、スクリーンリーダに対するバリアを軽減するにはどのような PDF ファイル作成方法が効果的か、の 2 点を中心に検証した。その検証結果をもとに、今後必要とされる電子フォーマットのアクセシビリティ関連の研究課題を考察した。

他方、視覚障害者の情報アクセシビリティにおいては、ソフト及びハードの利用環境が改善されても、その情報がユーザに伝わりにくいという問題もある^[8]。例えば、Web ページにおけるテーブルやフレームなどが有用な場合でも、音声ブラウザの当該操作が利用されていないがために、むしろ大変なバリアとして理解されることがある。スクリーンリーダによる PDF 文書の閲覧においても同様に、PDF 文書読み上げのための操作方法が周知でないために PDF 文書は全く読み上げられないとの誤解が生じることが考えられる。そこでユーザの利便のために、スクリーンリーダによる PDF 文書閲覧の留意点にも言及し、付録には国内スクリーンリーダの PDF 文書閲覧のための操作方法を示した。

2. アクセシビリティ検証の準備

PDF 文書のアクセシビリティ検証の条件は多岐にわたる。ここでは、Adobe 社がアクセシビリティのために MSAA¹ 対応を発表している Acrobat 5 以降と、その Acrobat 5 以降に対応しているとされるスクリーンリーダの各バージョンの組み合わせで検証を行った。

2.1 PDF 文書の元原稿

PDF 文書は一般に、Acrobat とは別のワープロ、DTP 等ソフトにより作成した文書を、Acrobat のアプリケーション

*1: 富士通株式会社

*2: 国立特殊教育総合研究所

*1: Fujitsu Co. Ltd.

*2: The National Institute of Special Education

1: MSAA (Microsoft Active Accessibility) : サードパーティによる Microsoft 製品へのアクセスシステムの開発を支援する技術体系。

ンであるPDF Maker等で変換して作成する。ここでは、PDF形式へ変換する文書として、現在広く使われているMicrosoft Wordで作成したWord文書、具体的にはヒューマンインタフェース学会が提供している論文用原稿テンプレートを使った^[12]。この文書には、タイトル・見出し・本文などの書式スタイルが明示的に定義されている。さらに、段組みが1段の箇所(タイトル等)と2段の箇所(本文等)にわかれているほか、図、表、テキストボックスなど一般的な印刷物の構成要素を含んでおり、アクセシビリティのチェック対象として適していると考えた。

2.2 PDF 文書の種類と検証条件

印刷原稿が同じとなるPDFファイルでも、その作成方法と設定により様々に異なった属性を有する。一般には、PDF文書をタグ付きとするかしないかで、スクリーンリーダーでのアクセスのしやすさが変わるとされている(Acrobat 5.0/6.0 オンラインヘルプ)。そこで、タグ付きとタグなし両方のPDF文書を作成する。

PDF文書の作成方法として、Word上のAcrobatマクロPDF Makerを利用するほかに、Wordのファイル印刷から仮想印刷ドライバAcrobat Distillerを選択する方法がある。PDF Makerでは変換時の設定でタグの有無を指定できるので両方を作成する。Acrobat Distillerでは常にタグが付かないので1種類のみ作成する。従って、タグの有無とPDFの作成方法により3種類に分かれる。

また、作成するAcrobatのバージョンとPDFのバージョンの違いがアクセシビリティに与える影響も検証する。AcrobatのバージョンはAcrobat 5とAcrobat 6の2種類である。本稿で取り扱うPDFのバージョンは1.4と1.5の2種類である(最新バージョンは1.5)。Acrobat 5では1.4を、Acrobat 6では1.4と1.5を作成する。従って、Acrobat及びPDFのバージョンの組み合わせは計3種類となる。

以上の組み合わせで計9種類のPDF文書を検証する。

3. PDF 文書のアクセシビリティの検証

3.1 文書閲覧のアクセシビリティ

視覚障害者にとって文書へのアクセシビリティとは、(1)文書の内容(テキスト)がスクリーンリーダーに送信されること、(2)キー操作により文書中の任意の箇所へ自在に移動できること、(3)キー操作により操作箇所のテキストが音声/点字で出力されること、と言える。

アクセシビリティのためには上記(1)が前提だが、

このとき、レイアウトをもった文書のテキストがどのような順序で連結されるかは、以後のスクリーンリーダーによる操作性(上記(2)と(3))に大きな影響を与える。そこで、PDF文書からスクリーンリーダーへ送信するテキスト情報の連結順序を調査した。調査では、Acrobatによるテキスト形式への保存状況と、スクリーンリーダーの全文読み上げ状況を検証した。

3.2 テキスト抽出の順序の結果

テキストファイルへの保存の方法として、Acrobat Reader 5の[テキストに書き出し]コマンドによるファイル保存と、Adobe Acrobat 6の[ファイル形式を指定して保存]による[Text(Plain)]と[テキスト(アクセシビリティ可能)]の計3通りで行った。

検証対象のPDF文書は9種類だが、実際にテキストファイルへ保存してみると、大まかな傾向としてAcrobat 6で作成したPDFのバージョン1.4と1.5については違いが現れなかった。また、PDF Makerによるタグなし文書とDistillerによるタグなし文書の間にも違いが現れなかった。従って9種類の区分から縮退して、Acrobat 5とAcrobat 6によって作成したそれぞれのタグ付きとタグなしの4種類が区分となった。これらテキスト保存方法とPDF文書の種類の組み合わせパターンは12種類となる。

12種類のテキストファイルの内容を確認すると、空行や空白が適度に入れてある違いを同一視すると大きく分けて5種類のパターンに分類された(表1)。さらに大きな分類を試みると、タグの有無により傾向が分かれた。Acrobat 5とAcrobat 6のどちらで作成するか、Acrobat 5とAcrobat 6のどちらでテキスト保存するかの組み合わせの間での大きな特徴の違いはなかった。

タグなしPDF文書からのテキスト保存では、画面に表示される印刷イメージを左から右、上から下へと読み進める順序にテキストが抽出された。一方タグ付きPDF文書では、Acrobat 5で作成したPDF文書をAcrobat 5で抽出したテキスト文書と、Acrobat 6で作成したPDF文書をAcrobat 6で抽出したテキスト文書が同じ傾向となった(表1中の値が3)。このテキストファイルは、テキスト本文での表組の挿入位置が不自然な場所になっている。印刷イメージを自然な順序に線型化したものでも、あるいは図や表組と言ったオブジェクトをテキスト化の末尾に追加するものでもない。元のWordファイルをWordの[下書き]表示で見ると、丁度同じ順序になっているこ

表1 PDF文書のテキスト保存の結果。表中の数字はテキスト連結順序の傾向を表しており、その詳細は表2に示した。

		Acrobat 5 で保存		Acrobat 6 で保存	
		テキストに書き出し		アクセシビリティ保存	Text(Plain)保存
Acrobat 5 で作成	タグなし	1'		1	2'
	タグ付き	3		4	3
Acrobat 6 で作成	タグなし	1		2	2
	タグ付き	5		3	3

とがわかる。つまり、元のファイルデータの順序を踏襲しているものと考えられる。

図や表組といったオブジェクトを含む文書の線型化の方法の妥当なものとして、

- (i) 印刷イメージを左から右、上から下に連結する方法、
 - (ii) 図や表組といったオブジェクトをページの冒頭あるいは末尾にまとめる方法、
 - (iii) 図や表組を、本文の中で参照を用いて言及されている、文脈的に適切と言える位置に挿入する方法、
- の3通りが考えられる。いくつかのパターンで(i)や(ii)に近い順序となったものがあつたが、いずれのパターンにおいてもここに掲げるような順序を乱れない形で実現しているものはなかつた。

3.3 スクリーンリーダでの読み上げの順序

次に、スクリーンリーダでの読み上げの順序を検証した。検証条件は、スクリーンリーダ3種類(PC-Talker-XP、

95Reader 5.0 (通称: XP Reader)、JAWS for Windows ver. 4.5) と Acrobat Reader 5、Adobe Reader 6 の組み合わせの計6種類である。ただしXP ReaderはAdobe Reader 6以降の対応とされているので、Acrobat Reader 5との組み合わせを差し引くと5種類となる。閲覧はWindows XP ProfessionalをOSとするPCで実行した。

スクリーンリーダへのテキストの渡され方として、Acrobat Reader 5では「読む順番の設定」の3種類の設定、Adobe Acrobat 6では「読み上げ順序」の3種類の設定がある。この各々でテキスト保存の検証で利用した4種類のPDF文書を読み上げさせた。

Acrobat Reader 5でタグ付き文書は「読む順番の設定」に影響されないとあるが、Acrobat 6で作成されたタグ付き文書には設定により読み上げの相違が現れた。これら22種類の読み上げ順序をまとめると表3のようになった。この結果を検討すると、PDF文書のタグの有無で傾向

表2 テキスト保存のパターンの内容
Table 2 Varieties of the text-exported files.

分類番号	1 ページ目	2 ページ目	3 ページ目	全体
1	整っている。	表1が脚注の前に、図1が脚注の後に飛ぶ。他の順序は整っている。	表2と表3が最後に来る。順序は整っている。	表が最後に飛んでいる。基本的に読み上げの順序としては整っていると言える。(ii)
1'	*2: Central...が消失。	Fig...の文字列が消失。		
2	著者所属が3.本文の前に飛ぶ。	順序は整っている。図1の内容文字列、キャプションが消失。	整っている。	1ページ目で著者所属が3.の前に飛ぶ意外は、見た目通りの順序になっていた。(i)
2'		図1が消失。		
3	著者所属が冒頭に来る。順序は整っている。	図1が最後に飛ぶ。脚注が本文6.に埋め込まれる。	表2が8.1の間に、表3が謝辞の間に挟まる。	1ページ目、2ページ目はほぼ整っていた。3ページ目は表の位置が飛んでいた。元のWordファイルの「下書き」で表示した順序そのままである。
3'		図1が消失。		
4	整っている。	表1が冒頭に、図1が7.の前に飛ぶ。	表2が9.の間に挟まる。表3が最後に飛ぶ。	1ページ目は整っているが、2ページ目3ページ目は表が文章の間に挟まっていた。
5	*2: Central...が消失。順序は整っている。	表1が冒頭に、図1が最後に飛ぶ。脚注が消失。	9.LaTeXの節が冒頭に、その見出しのすぐ後に表2が挟まる。謝辞の間に表3が挟まる。	1ページ目、2ページ目はほぼ問題ない。3ページ目は9.節が冒頭に来ると言う順序の乱れがあつた。
6	3.1と3.の見出しの順序が入れ替わっている。*2: Central...が消失。	5.1の見出しが図1の見出しの前に飛ぶ。	8.1節の見出し、表2の見出し、8.節の並びで順序が乱れる。	大きな乱れはないが、節見出しと言う重要な行の位置が飛ぶ問題があつた。(i)
7	はげしく順序が乱れる。	4.3が冒頭に飛ぶ。	表が縦に連結される。表の右列は表3と表2で連結される。	1ページ目は理解不可能なほどに順序が崩れた。3ページ目も順序がやや乱れると共に、表が縦に連結されてしまった。
8	*2: Central...が消失。順序は整っている。	表1が冒頭に、図1が末尾に飛ぶ。	9.LaTeXの間に表2が挟まる。謝辞の間に表3が挟まる。	3ページ目で表の位置が飛んでいた。
9	整っている。	表1は印刷通り6.2の前に入る。図1が脚注の後に飛ぶ。	表2と表3が最後に来る。順序は整っている。	問題なく線型化された。2ページ目表1の位置は印刷通りだった。
10	整っている。	表1が冒頭に、図1が最後に飛ぶ。	9.LaTeXの見出しが冒頭に飛ぶ。表3が謝辞の中に挟まる。	3ページ目で表の位置が飛んでいた。
11	*2: Central...が消失。順序は整っている。	表1が冒頭に飛ぶ。図1が7.の直前に飛ぶ。	9.LaTeXの間に表2が挟まる。表3が最後に飛ぶ。	2ページ目、3ページ目で表の位置が飛んでいた。
12	2.1見出しが先頭に飛ぶ。3.1と3.の見出しの順序が入れ替わっている。*2: Central...が消失。	見出し行の位置がかなり乱れている。	8.1節の見出し、9.の見出し、8.節の並びで順序が乱れる。	見出し行の位置が大きく乱れる問題があつた。

表3 スクリーンリーダによる読み上げ順序の結果．表中の数字については表2を参照のこと．

Table 3 Result of text sequences by screen readers.

		Acrobat 5			Acrobat 6		
		テキストの読み込み順序を自動的に選択	標準の読み込み順序を使用	左から右へ、上から下へ	文書の読み上げ順序を支持	左から右、上から下への順序で読む	印刷ストリーミングの読み上げ順序を使用
Acrobat 5	タグなし	7	1	6	2	9	1
で作成	タグ付き	3'	-	-	3	10	11
Acrobat 6	タグなし	7	1	6	2	9	1
で作成	タグ付き	7	8	12	3	10	11

を大きく分類できる．タグなしの文書は，Acrobat Reader 5 の環境で読ませると順序が大きく乱れた(図1)．Acrobat 6 で作成したタグ付き文書を Acrobat Reader 5 の環境で読み上げさせると順序が大きく乱れた．Acrobat Reader 5 の [読む順番の設定] を変更すると若干の乱れが残るものの順序がかなり改善された．

Acrobat Reader 5 の [読み上げの順序] を [標準の読み込み順序を使用] に変更すると，図や表組がページの末尾に配置される方法で整った順序に読み上げられた．Adobe Acrobat 6 の環境ではほぼ問題ない順序で読み上げられた．

Acrobat Reader 5 において，PC-Talker-XP と JAWS の読み上げ順序の違いはなかった．また，Adobe Acrobat 6.0 においてそれぞれのスクリーンリーダでの読み上げ順序の違いはなかった．これは内部的に同一の手続きで Acrobat Reader からテキスト情報の受け渡しをしていることによると思われる．

これ以外の Acrobat Reader / Adobe Reader の環境とタグ付き文書の組み合わせの読み上げは前節のテキスト抽

出で検証したものと同じく元の Word のデータの順序で読み上げられた．

タグなし文書については，Acrobat Reader / Adobe Reader の読み上げの順序の設定を変更することにより読み上げ順序の乱れを改善できた．

いずれの組み合わせにおいても，前節で掲げた(i)から(iii)のような順序を乱れなく実現しているものではなく，ほぼ見た目の通りの読み上げとなった．

3.4 スクリーンリーダでの文書閲覧の操作

スクリーンリーダによる実用的な文書閲覧のためには，ページより細かい単位での操作・読み上げ機能が必要である．例えば，スクリーンリーダによるワープロ文書閲覧では，カーソル操作に対する読み上げが一般的である．一方 PDF 文書は元来編集を目的としないので Acrobat Reader / Adobe Reader は特定の1文字にフォーカスを当てるような仕様を有していない．このため，各スクリーンリーダは独自でこのような機能を実現しなければならない．

また，Acrobat Reader / Adobe Reader では画面表示を目

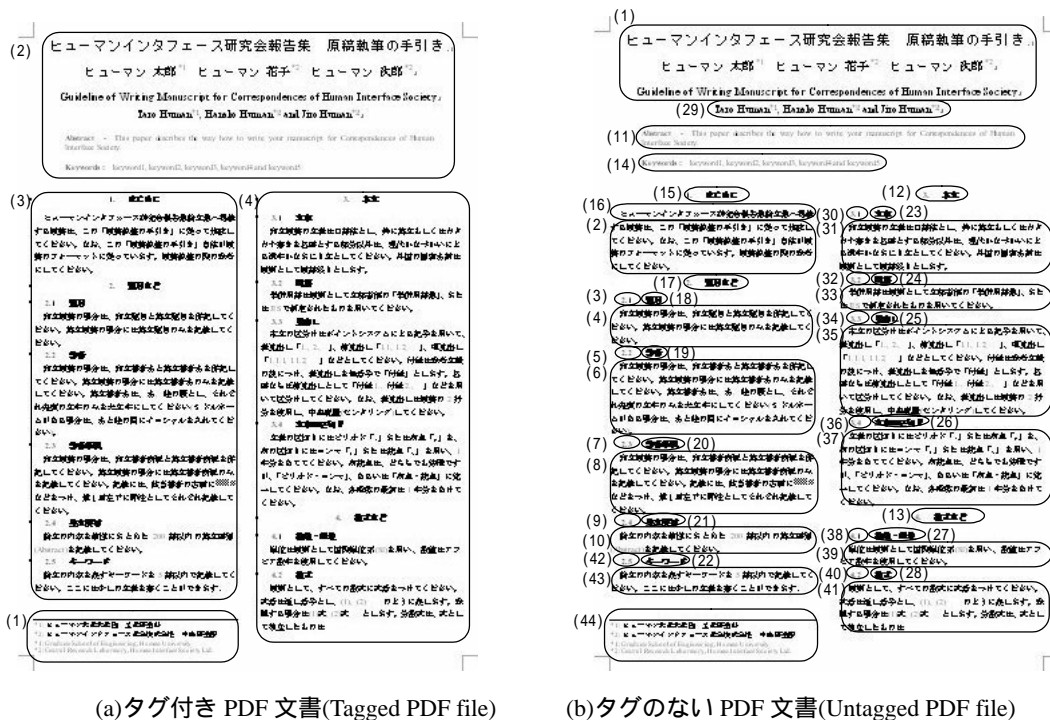


図1 PDF 文書のタグの有無によるページ内テキストの読み上げ順序の違い．番号が読み上げ順序を表す．
Fig.1 Difference of the order of reading a whole page.

的としてページ移動などのキー割り当てをもつ。ページ単位や表示画面と言った物理的単位や、「しおり」といった文書の論理的単位のジャンプである。

ここでは、スクリーンリーダ独自のナビゲーション操作とそのときの読み上げ、及び Acrobat Reader / Adobe Reader 標準のキー操作に対する読み上げを検証した。さらに、表示中のページ数のように、現在の閲覧状態を自動的にあるいはインタラクティブに知らせる機能の実装も検証項目に加えた。

PC-Talker-XP と JAWS の Acrobat Reader 5 と Adobe Reader 6 での動作の違いは、Adobe Reader 6 では動作が大変遅くなる現象が観察された以外に機能面ではほぼ同じであった。そこで以下では、各スクリーンリーダごとに要点をまとめる。

(1) PC-Talker-XP

PC-Talker-XP の独自キー操作により全文読みできた。行、文字の読みを実現する仮想カーソル機能はなかった。

Acrobat Reader 標準の前後ページへの移動機能では、遷移先のページ数を読み上げた後にページの第 1 段落を読み上げた。任意ページ移動機能も異動先の第 1 段落を読み上げた。PC-Talker-XP のページ数読み上げコマンドは常に「1 ページ」と読み上げていた。

「しおり」によるナビゲーションについては、「しおり」パレットでの上下矢印キーによる読み上げが意味不明の文字列になっていた。「しおり」により移動すると該当のページを表示し第 1 段落を読み上げた PC-Talker-XP には読み上げ位置に相当する仮想カーソルが存在せず、読み上げはページ単位の動作であり、複数のしおりが単一ページに対して付与されていても、「しおり」に相当する段落からの読み上げを実現することはできない。

Adobe Acrobat 6 では動作が大変重たく実用的に利用できなかった。「読み上げ順序」の設定をデフォルトの「文書の読み上げ順序を支持」から「左から右、上から下への順序で読む」あるいは「印刷ストリーミングの読み上げ順序を使用」に変更すると動作が遅い問題が解決された。しかし PC-Talker-XP の全文読みコマンドが機能しなくなった。Esc キーを押す、あるいは Acrobat Reader のウィンドウを非アクティブにし再度アクティブにすると全文読みを行うという現象が確認された。

(2) JAWS 4.5

PDF 文書中で仮想的なカーソル（画面上は非表示）を動かすナビゲーション・カーソル機能がある。このカーソルを上下左右の方向キーで動かすと、行読みと 1 文字読みが可能だった。

Acrobat Reader 標準の前後ページ移動及び任意ページ移動機能により任意のページへ移動すると、異動先ページの全文読みを行いナビゲーション・カーソルも移動先ページの先頭に移動した。

「しおり」のナビゲーションについて手元の環境では、

画面表示は変化するもののナビゲーション・カーソルのフォーカスが該当の文書位置に移動しなかった。「しおり」パレットの上下矢印キーの操作においてラベル文字列の正しい読み上げに先立って余計な文字列が読み上げられる不具合がしばしば発生した。「しおり」ツリーの階層の深さ情報は音声化されていなかった。

Adobe Reader 6 では動作が大変遅く実用的に利用できなかった。PC-Talker-XP と同様に「読み上げ順序」の設定をデフォルトから変更すると、この問題は解決された。

(3) XP Reader

テキストウィンドウという機能で代替的に仮想カーソル機能を実現している。そのウィンドウ内では、通常のエディタと同じ要領で、上下左右の方向キー、(Ctrl+) Home / End キーを使ってカーソルを移動させることで、行読み、1 文字読みができた。

Acrobat Reader 標準の前後ページ移動及び任意ページ移動機能により移動先のページを自動的に読み上げた。XP Reader のページ数読み上げコマンドで表示中のページの確認ができた。

「しおり」については、「しおり」ラベルの内容を正しく読み上げた。しかし、「しおり」ツリーの階層の深さ情報は音声化されなかった。PC-Talker-XP と同様、仮想カーソルが存在しないため、読み上げの基点は表示ページの冒頭となり、同一ページの途中で「しおり」が付加されていても活用することができなかった。

4. アクセシブルな PDF ファイルの作成法

4.1 タグ付き PDF

タグ付き PDF 文書は、表示のための PostScript データのほかに、文書に付随する構造情報を有する。その情報を頼りにスクリーンリーダが読み上げの処理を実現する。

Word ファイルからタグ付き PDF 文書を作成すること自体は難しくない。PDF Maker の PDF 変換設定のダイアログボックス（実際には、Word のメニューバーに追加されている）でタグ付きを指定すれば、あとは Word タスクバー内のアイコンあるいは Adobe PDF メニューの [Adobe PDF への変換] を操作することで、自動的にタグ付き PDF 文書を作成することができる。

Word 上から PDF 文書を作成する方法としてもう 1 つ、仮想プリンタドライバへ印刷する方法がある。この場合は PDF Maker が用いられず、直接 PDF 生成をする Adobe Distiller が処理を開始する。この方法では生成される PDF ファイルにタグが付加されない。それに伴って Word のスタイル情報も引き継がれないので、「しおり」の自動生成も機能しない。この意味で、Microsoft Office 系から PDF を作成する場合、PDF Maker を利用することが望ましい。

4.2 論理構造の自動生成

PDF 文書は、Word / 一太郎などのワープロ文書を変換して作成する。その際、論理構造に関する情報も受け継

ぐ。例えば、Word 文書の「見出し」スタイルを PDF 文書の「しおり」に変換することができる。従って、変換前の文書の作成者には、文書内の書式をスタイル（この用語は Word の場合、フォント、サイズ、行揃えなどを規定する）で定義して、これを論理的な構造と対応させることが求められる。

4.3 アクセシブルな PDF 文書

Acrobat 5 / Acrobat 6 Professional を使えば、文書作成者がタグを編集し直すこともできる。また、レイアウト表現を自動線型化したことにより崩れた文書の順序を訂正することができる。

テキスト抽出及びスクリーンリーダーの読み上げの検証で確認されたように、テキストの順序の自動判定はまだ不十分である。アクセシビリティ向上のためには、手間はかかるが、タグの内容を人が確認した方がよい。

4.4 PDF 文書のセキュリティ設定

PDF 文書のセキュリティ機能によりスクリーンリーダーへのテキスト送信を禁止することはアクセシビリティを完全に無効としてしまうことを確認した。また、テキスト抽出を禁止しつつスクリーンリーダーのアクセスを可能とする設定が両立することも確認した。公開する PDF 文書では、スクリーンリーダーへのアクセスの許可の確認が望まれる。

5. PDF 文書閲覧のためのユーザ側の設定

Acrobat Reader / Adobe Reader の標準設定では、10 ページ以下の PDF ファイルをスクリーンリーダーで読み上げさせるとページ区切りが機能せず、閲覧性が良くないばかりか場合によっては動作が大変遅くなったり、不安定になったりする現象が発生した。

確実に 1 ページずつ処理させるためには設定の変更をする必要がある。これは、Adobe Reader / Acrobat Reader の編集メニューの環境設定を開き、アクセシビリティの設定の [長い文書の場合、アクセシビリティインタフェイスに 1 ページずつデータを配信するよう選択できます] の設定を 1 ページ以下にすることである。

6. 考察

6.1 PDF 読み上げの検証結果について

PDF 文書の種類とそこからのテキスト抽出方法の種類との組み合わせにおいて、いずれも完璧な線型順序での抽出はできなかった。スクリーンリーダーユーザにとって適切な順序を実現するためには手作業も必要と思われる。

スクリーンリーダーによる PDF 文書の読み上げの順序はスクリーンリーダーの種類には依存せず、テキスト抽出の順序と同じであった。また、Adobe Reader 6 と PC-Talker-XP あるいは JAWS の組み合わせは動作が大変鈍く不安定で実用的ではなかった。

6.2 読み上げ仮想カーソル機能

既に述べたとおり、音声読み上げや点字表示といった同時的表示箇所が制限される表示媒体においては、文書閲覧時に該当箇所を指し示し自由に操作可能な読み上げ仮想カーソル機能が求められる。テキストファイルをエディタないしワープロで閲覧する時の実際のカーソルに相当するものである。エディタやワープロの実際のカーソルを操作することにより行、文字単位の読み上げが可能となる。視覚障害者の文書閲覧フォーマットとしてテキストファイルが好まれる大きな要因はこの点にある。

JAWS 4.5 はナビゲーションカーソルと名付けられた操作で行、文字単位の移動と読み上げができた。XP Reader で行、文字単位の閲覧をするためにはテキストウインドウを開く必要がある。PC-Talker-XP で行、文字単位の読み上げを行うためには、PC-Talker-XP 独自の機能であるクリップボード内の仮想カーソル機能であるクリップボードカーソルを応用的に利用する必要がある。これは、PC-Talker-XP において Windows のヘルプファイルを閲覧するときの手法である。

このように読み上げ仮想カーソルの意味において、ファイルフォーマットを意識することなく自然な閲覧を実現しているのは JAWS 4.5 と言える。他の 2 つのスクリーンリーダーの方法は代替的に読み上げ仮想カーソルを実現しているにとどまる。この方法では文書中にリンクやフォームフィールドなどのオブジェクトが登場した際に読み上げ位置とフォーカス位置の整合性に問題を抱えることになると考えられる。

6.3 文書ナビゲーション機能

視覚障害者が文書を自在に扱うには、読み上げ仮想カーソルを文書中の意味のある位置へ即座に移動させるナビゲーション機能が求められる。その重要性は、晴眼者が文書を閲覧する時に、まずページ内のレイアウトから骨格部分を見だし、行空けや字下げ大きな太字により章節区切りを確認して文章の空間的な構造把握をする状況から想像すれば自明であろう。

ナビゲーションの単位は、文書の物理的な構造では、ページ、段組があれば段、論理構造で見た場合、章、節、項の見出し、段落を単位として使えば便利である。

HTML 閲覧時を例とするならば、物理的な構造を利用した機能として、PageUp キーや PageDown キーによる 1 画面分や数十オブジェクトごとのスキップ、論理的な構造を利用した機能として見出しタグごとの見出しスキップ機能が挙げられる。

PDF 文書閲覧においては、いずれのスクリーンリーダーでも、PDF 文書の標準の物理的な区切りであるページについて Acrobat Reader / Adobe Reader の標準のコマンドキーで操作することができた。移動後のページ自動読み上げについては、JAWS と XP Reader は移動後のページを、PC-Talker は移動後のページ数を読み上げた後、表示ペー

ジの第1段落だけを読み上げた。

PDF文書の有力な論理的な構造として「しおり」が挙げられるが、スクリーンリーダーの「しおり」のナビゲーション機能は、「しおり」パレットのラベルを正しく読み上げない、ツリーの階層構造が音声で表現されていない、「しおり」の飛び先を正しく読み上げないなど、機能の対応にはまだいずれも改善の余地があることがわかった。

HTML文書の閲覧の場合を参考にするならば、文字読み、行読み、ヘッダー読み、リンク読み、テーブル読みなどができるホームページ・リーダー^[13]と比較しても機能が少ない。これら読み上げ仮想カーソル、ナビゲーション機能の改善が既存のスクリーンリーダーには求められる。HTML音声対応の黎明期はスクリーンリーダーの一般ブラウザ対応への負担からHTML閲覧専用の音声ブラウザが開発された。専用ソフトは音声という提示形式、キーボードによるナビゲーションを前提として適切な仕様が検討されている。これに習ってPDF文書に特化したPDF閲覧音声点字ブラウザが開発されてもよい。

6.4 論理構造化された電子文書

本編で確認した通り、PDF文書を構造化することは半自動化されているが、レイアウトの順序の手直しなどよりアクセシブルな論理構造化をする作業は大変な工数を必要とする。しかし、テキストの順序、「しおり」といった構造はスクリーンリーダーによる閲覧において大変重要な留意点である。Word文書の「見出し」スタイルをPDF文書の「しおり」に自動変換することができる。今後、公共的な機関で作成するPDF文書は、最低限の論理構造、タグ付きとなっていることを保障することが望まれる。

しかし一般的には、スタイル設定のような技巧的な方法は仕様せず、スタイルは「標準」のまま、文字のフォントやサイズを書式で変更して視覚的に「見出し」として利用してしまっていることが多いと思われる。この問題は文書のPDF化時だけの問題ではなく、普遍性がある。HTML文書の見出し記述において、見出しタグを使用せずに文字寸法を変更することにより視覚的に見出しを描写することで、読み上げブラウザの見出しナビゲーションが機能しないという問題がある。つまりこの問題の本質は、「見出し」という論理構造を機械的に処理可能とする内部的な情報が存在しないということである。

全く同様の原因で、例えばワープロ文書を自動点訳する際にも同種の問題が生じている。村田と池田の点字出力システムでは、HTMLのタグを処理することで点字文書の構造化を図っている^[14]。しかし、処理対象の電子文書は視覚的装飾により構造化を擬似的に表象しており、システム実用化のためには、これらを自動的に構造化処理することが課題となっている。つまり、スタイルが規定されていないワープロ/DTP文書を、文書内の文字の書式や配置などの情報をもとに、経験的な手法で自動的に構造化していくシステム(ソフトウェア)の開発が求

められる。

この実装に当たって、中間媒体となる文書の書式をどれにするかは、別途検討する必要があるだろう。昨今Microsoft社のOffice 11、石川准氏開発の自動点訳ソフトEXTRAやその他OCRソフトの保存形式等、視覚障害者が利用する可能性の高いアプリケーションにもXML対応が進められている背景を考慮すれば、XMLにより文書を正しく構造化することが有力な手段として考えられる。その後スクリーンリーダー対応が充分に行われているアクセシブルなフォーマットや点字形式などに再変換すればよい。このような手順を可能とする文書論理化システムがあれば、視覚障害者にとって電子文書全般へのアクセシビリティの向上が期待できる。

7. まとめ

PDF文書のアクセシビリティを実現するために、データの変換性という点でPDF文書のテキスト形式への保存を検証した。文書レイアウトの線型化を保証する構造化の自動化にはまだ課題が残っていることがわかった。

スクリーンリーダーへのPDF文書のアクセシビリティを検証したところ、検証時点(平成15年6月~10月)では改善が進んでおり、3種のスクリーンリーダーでPDF文書を音声化可能だった。しかし、文書内での操作性にはまだ多くの課題が残されていた。

第4章ではよりアクセシブルなPDFの作成法について言及した。今回の検証から、アクセシブルにする解決手段の煩雑性や背景の原理はHTMLのアクセシビリティ問題に類似することがわかった。この類似性からPDF文書のアクセシビリティ問題の解決の困難性が想像される。しかしその作業の背景にある手法や作業に向けての取り組み方はHTMLのそれを手本にできると考えられる。

PDF文書のアクセシビリティ問題の考察から、一般的に情報環境のアクセシビリティを改善するためには、支援技術製品メーカーだけでなく、OSメーカー、文書/コンテンツ作成者、そしてWebアクセシビリティの研究者等の共同作業が必要であると考えられる。

謝辞

本検証作業の一部は、科学研究費補助金(基盤研究(A)(2)14208020)によります。

参考文献

- [1] Web Content Accessibility Guideline 1.0, <http://www.w3.org/TR/WAI-WEBCONTENT/>
- [2] M. G. Paciello: Web・アクセシビリティ すべての人に優しいWeb・デザイン, アスキー (2002).
- [3] Asakawa, C. and Takagi, H.: Annotation-based transcoding for nonvisual Web access, Proceedings of ACM ASSETS 2000, pp.172-179 (2000).
- [4] 斎藤照花, 堀内靖雄, 市川薫: インタラクティブ性を考慮

- したウェブアクセシビリティの可能性の検討, 信学技報 WIT2001-29 (2001).
- [5] 藤原敦史, 渡辺実, 堀内靖雄, 市川薫: 表及び数式の音声化の検討, ヒューマンインタフェースシンポジウム論文集, pp.643-648 (1997).
- [6] 石田優子: Web・ユーザビリティ&アクセシビリティ・ガイドライン, 毎日コミュニケーションズ (2003).
- [7] 石川准: Web アクセシビリティの現状評価と将来への期待, <http://fuji.u-shizuoka-ken.ac.jp/~ishikawa/web.ppt>
- [8] 渡辺哲也, 指田忠司, 長岡英司, 岡田伸一: 視覚障害者の Windows パソコン及びインターネット利用・学習状況, 信学技報, WIT2002-62 (2002).
- [9] 総務省, 平成 14 年版情報通信白書, ぎょうせい (2002).
- [10] 国立印刷所: インターネット版「官報」ご利用案内, <http://kanpou.npb.go.jp/>
- [11] 村田拓司: 視覚障害者の IT 時代: 福祉労働, No.92, pp. 48-56 (2001).
- [12] ヒューマンインタフェース研究会報告集 原稿執筆の手引き, <http://www.his.gr.jp/download/meeting/meeting.doc>
- [13] C. Asakawa and T. Itoh: User Interface of a Home Page Reader, Proceedings of ACM ASSETS '98, pp.149-156 (1998).
- [14] 村田健史, 池田尚志: 遠隔地からの点字出力システム, 信学技報, WIT2003-5 (2003).

付録：各スクリーンリーダの PDF 文書への対応状況

PDF 文書内での各スクリーンリーダの読み上げ/ナビゲーション機能とキー割り当て, 及び実際に使用した状況を以下に記す. 箇条書き記号の はスクリーンリーダ独自, は Acrobat Reader 標準のキー操作を表す.

(1) PC-Talker-XP + Acrobat Reader 5.0

ページ読み (全文読み) (Shift キーまたは Ctrl+Alt+A) ...画面に表示されているページを全文読み上げる
いずれのページを表示中でも, ページごとの読み上げではなく, 文書の全文読みが行われた.

スクロール読み (上下矢印キー) ...画面の上部に表示されている段落を読み上げる

ページの最上部の段落を音声で読み上げた.

ページの内容をクリップボードにコピー (Ctrl+Alt+S)

コピー内容は, 全文読みによる読み上げと同じで, Acrobat 5 で作成したタグ付き PDF 文書は論理構造を保っていたが, これ以外の文書では構造が崩れていた.

ページ番号読み (Ctrl+Alt+F9)

どのページを閲覧中に実行しても, 「いちページ」と読み上げられた.

前/後のページへ移動 (左/右方向キー, Ctrl+上下方向キー, Ctrl+PageUp / PageDown)

移動先ページの第 1 段落を読み上げた.

開始 / 最終ページに移動 (Ctrl+Shift+PageUp / PageDown)

移動先ページの第 1 段落を読み上げた.

(2) JAWS + Acrobat Reader 5.0

行読み (上下方向キー) ...仮想的なナビゲーション・カーソルを上下に 1 行移動させる

移動先の 1 行を読み上げた.

1 文字読み (左右方向キー) ...仮想的なナビゲーション・カーソルを左右に 1 文字移動させる

移動先の 1 文字を読み上げた.

しおり間移動 (Ctrl+上下方向キー) ...仮想的なナビゲーション・カーソルを「しおり」の最上階層の間で移動させる

動作環境により, 移動先より下のテキストを読み上げた場合と, 読み上げなかった場合 (本文で記述) とがあった.

文書の先頭行 / 最終行へ移動 (Ctrl+End / Home) ...仮想的なナビゲーション・カーソルを文書の先頭行 / 最終行へ移動させる

移動先の 1 行を読み上げた.

前/後のページへ移動 (Ctrl+PageUp / PageDown)

移動先ページの先頭へナビゲーション・カーソルが移動した.

ページ番号を指定して移動 (Ctrl+N)

移動先ページの先頭へナビゲーション・カーソルが移動した.

開始 / 最終ページに移動 (Ctrl+Shift+PageUp / PageDown)

移動先ページの先頭へナビゲーション・カーソルは移動しなかった.

(3) XP Reader 5.0 + Adobe Reader 6.0

以下のスクリーンリーダ独自機能はすべて操作名通りに動作した.

テキストウィンドウを開く (Ctrl+Alt+V) ...PDF 文書の全テキスト情報を抽出したテキストウィンドウを開く

テキストウィンドウ内の 1 行読み (上 / 下矢印キー)

テキストウィンドウ内の 1 文字読み (左 / 右矢印キー)

テキストウィンドウ内の行頭 / 行末へ移動 (Home / End)

移動先で 1 文字読みを行った.

テキストウィンドウ内の文書の始め / 終わりへ移動 (Ctrl + Home / End)

移動先で行読みを行った.

前/後のページへ移動 (左/右方向キー, Ctrl+上下方向キー, Ctrl+PageUp / PageDown: Acrobat Reader 標準), 開始 / 最終ページに移動 (Ctrl+Shift+PageUp / PageDown)

1 ページ目へ移動した際に文書の先頭から読むのはよいが, 2 ページ目へ移動したときも 1 ページ目の内容から読み始めたり, 3 ページ目へ移動したときは何も読まないなど, 適切に動作しているとは言えなかった.