

August 31, 2009

RT0874

Human-Computer Interaction 12 pages

Research Report

Automatic draw method of a reading assist ruler using OCR technology

Hiroaki Ozeki, Takuyra Ohko, Takashi Saitoh, Shinji Iizuka

IBM Research, Tokyo Research Laboratory

IBM Japan, Ltd.

1623-14 Shimotsuruma, Yamato

Kanagawa 242-8502, Japan



Research Division

Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Limited Distribution Notice

This report has been submitted for publication outside of IBM and will be probably copyrighted if accepted. It has been issued as a Research Report for early dissemination of its contents. In view of the expected transfer of copyright to an outside publisher, its distribution outside IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or copies of the article legally obtained (for example, by payment of royalties).

Overview of the idea:

Senior citizens and people with learning disabilities have difficulties in reading long sentences spanning multiple lines. Sometimes they lose track of the current line or they have to make a special effort to keep their eyes on the proper line. Even people without such disabilities may lose track, especially when doing something else concurrently, reading complex sentences or reading in bad shape. To prevent losing track of lines, people sometimes move their fingers on the PC display or invert the sentence by highlighting it with the mouse device. However, there are few IT assistive tools to address the problem.

One of the existing methods developed is to display a ruler (a horizontal bar) on the PC screen for separating the lines, as provided in IBM Easy Web Browsing, or the ScreenRuler of Claro Software Inc. Senior citizens, however, are often uncomfortable about controlling the ruler itself, since they must use the mouse to adjust the ruler position at each line of text. Also, people with learning disabilities have difficulties in finding the space between the lines as a target for positioning the ruler. Considering these problems, the current rulers do not seem to be perfect solutions.

The first disclosed is a method of precise and fast analysis of sentence structures of Web content using a combined method of partial OCR and HTML content analysis. The second is a method of obtaining a candidate list of ruler positions by using the sentence structures. The third is a method of moving a ruler appropriately for the purpose of silent reading by using the candidate list. The fourth is a method of automatically determining the ruler attributes (width, color, line type) and its shape according to the mouse position located by a user.

Background(課題、従来技術の問題点)

【課題】

複数行に渡る文章を読み進めるとき、学習障害や高齢者によっては、行を読み違えることがある。あるいは、読み間違えないように過度に意識する必要がある。健常者でも、読んでいる途中で別の作業をするような場合、読み込んだ文章を読むような場合、あるいは体調のすぐれない時には、同様の問題に遭遇することがある。こうしたとき、画面に指先を当てて読み進めたり、マウスで選択状態にして読み進めるなどの方法が取られることがあるが、専用の IT による支援ツールはあまり提供されていない。

【従来技術の問題点】

支援方法の一つに画面にルーラー(横棒)を表示する方法がある。IBM の EWB や claro software 社の ScreenRuler (または通常の Web ブラウザ)が既存技術としては存在するが、高齢者においては、ルーラーの箇所を示すためのマウスの細かい指示が苦手であったり、学習障害者においては、行間を見つけること自体が困難でルーラーをうまく活用できないというジレンマがある。

Summary of Invention(発明の要点、解決法)

【発明の要点と解決法】

- 部分 OCR とコンテンツのソース(文書フォーマット(HTML 等))レベルの解析を用いた、表示コンテンツの文章構造の高速かつ正確な把握処理。(部分 OCR は限定辞書により辞書データサイズを抑える。)
- 判別された文書構造に基づいたルーラー配置位置候補リストの取得。
- 黙読の支援を前提とした、配置位置候補リストに基づく、ルーラーの適切な移動に関する処理。
- ルーラーが配置される位置に従った、ルーラーの適切な属性(太さ、色、線種、文字への重なり回避)を自動的に決定する方法。

【効果】

- 従来技術の問題点を解決し、課題の解決に貢献する。
- 画像中の文字、動画中の字幕(ex. 字幕部分の囲み強調)、Flash にも有効な手法。
- ヘッダー、ナビゲーションや段組といった細かいレイアウトがあっても、適切にルーラーを移動可能。
- EWB 特有の機能「ルーラ」「ハイライト」「分かち書き」「縦書き表示」「音声読み上げ」を強化できる。
- 自動的に縦書きと横書きのルーラーに対応できる。
- HTML コンテンツに限らず様々なフォーマットの文書に対応可能。(MS アプリ、Notes など対応化)
- 本アイデアにより、画像中の文字も取得されているので、それを読み上げることが可能になる。
- ルーラーと同様の目的を持った、マスキング、ハイライトといった機能にも活用できる。

※ HTMLソースから取得した位置情報が利用者(Web 閲覧者)の画面表示の直感と異なるケース。

- ・画像上に描かれた文字、Flash 広告などは、ブラウザからは IMAGE や OBJECT として認識されるが、ユーザは画面表示の一部と認識している。
- ・高度なデザインを実現するために CSS のオーバーレイ(重ね合わせ)表示が多様されるようになってきている。複数の HTML Contents が重ね合わせて表示されており、IE の JavaScript 関数で周辺文字列を取得した情報と、実際に Web ブラウザで最前面に表示されている情報が異なる場合がある。
- ・JavaScript による動的な変更、アニメーション処理により、利用者の見る情報がユーザ操作や時刻で変化することがある。

Description (発明の全体像、データ処理手順、GUI、効果)

1. ルーラーに対する移動の指示があった時に、そのルーラーを、部分的な OCR 結果と、対象となるコンテンツをソースレベルで解析した結果の、両方を適宜利用してルーラーの配置位置を算出することで、常に利用者の望む位置にルーラーを移動する方法。
可能になること
 - ・常に行の下にルーラーを配置する。
 - ・n 個下の行と指示があれば、n 行下の行にルーラーを移動する。
 - ・2 段組などで、一つ目の段から二つ目の段に移動するような場合、最下部に到達したルーラーを最上部に移動する。
 - ・Flash などソースレベルの解析が困難なコンテンツであっても、あるいは、画像に埋め込まれたようなテキストであっても、それに対して、ルーラーを配置する。
2. ルーラー配置の指示のあった場所を中心として、部分的な OCR と、コンテンツのソースレベルの解析を行い、そのどちらを用いるかを両者の結果の親密度から決定(6)し、ルーラーの配置候補リストを作成する方法。
3. ルーラーの移動の指示があったとき、配置候補リストから移動先候補を得る方法。
4. 移動先候補が配置候補リストにない場合、再び 2 を行って配置候補リストを作成する方法。
5. ルーラーをドラッグするなどして、ルーラーを配置すべきあたりを取得する方法。
6. 二つの配置候補リストの中心にある文字列を比較し、それが等しければコンテンツのソースレベルの解析結果が優位であるとし、閾値以上に誤りが多い場合は、部分 OCR 結果が優位であるとする方法。

7. 3 によって得られた移動先候補にルーラーを引くとき、その対象となる文字列に従って、ルーラーの線種、太さ、色を変える方法、および、ルーラーと文字が重なる可能性がある部分を回避する方法。

* この他、HTML レベルでルーラーを追加する方法もあるが、このレポートには含めない。

システム全体の処理・状態遷移図

- A: ルーラーの表示・非表示制御。配置位置リストと配置位置オフセットに従ってルーラーを表示。
- B: ルーラーをホールドして任意に移動する処理。行間など関係なく、自由に移動。ホールドが解除されると、その時のマウスポインタ位置を頼りに付近を部分 OCR などして文書構造を解析。配置位置リストを再作成および配置位置オフセットをリセット。
- C: ルーラーをカーソルなどコマンドでステップごとに移動する処理。移動後の候補位置が配置位置リストの中にある場合は、それを指すオフセットに従う。もし、配置位置リストから出てしまった場合は、越える直前の配置候補位置を基準に、部分 OCR などして文書構造を解析。配置位置リストを再作成および配置位置オフセットをリセット。

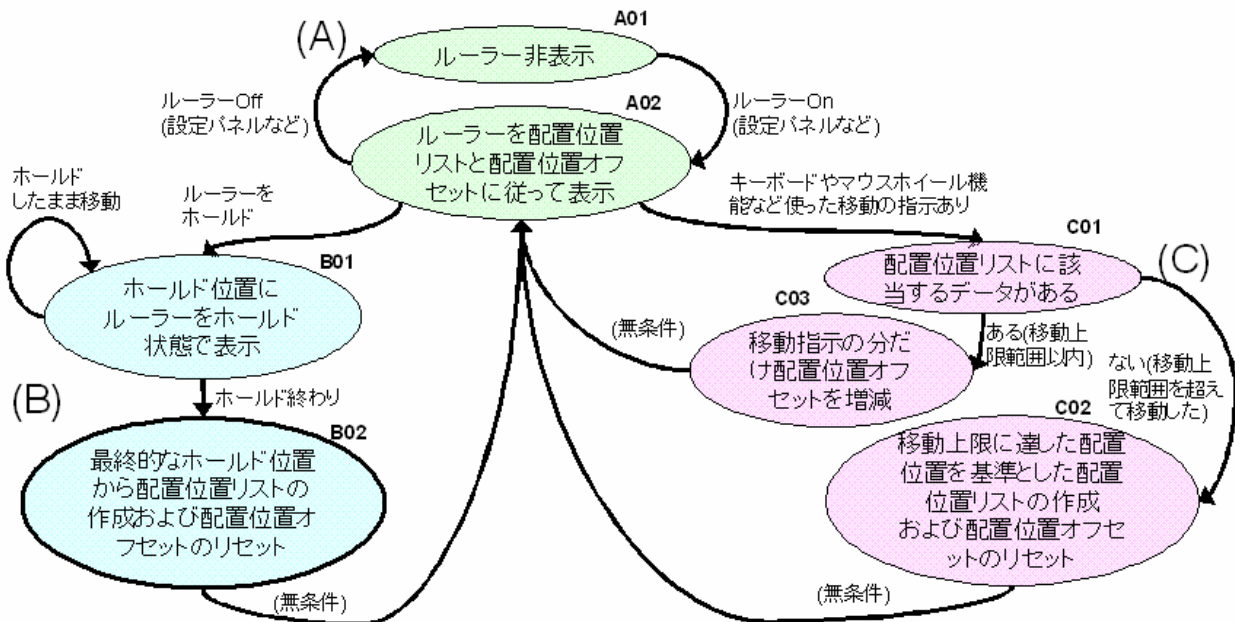


図 1 状態遷移図

A01 および A02

A01

ルーラーの表示をやめる。
ルーラーの表示の指示があった場合は
A02 に移動する。

A02

配置位置リストの中から、配置位置オフセットで示される表示方向および表示位置に従ってルーラーを表示する。
ルーラーのホールドを検知したら B01 に移動する。
ルーラーの移動指示を検知したら、C01 に移動する。

配置位置リストのデータセット例

(size, zp, (ori_0, posx_0, posy_0),(ori_1, posx_1, posy_1),(ori_2, posx_2, posy_2), ... (oni_n, posx_n, posy_n))

size: 配置候補データ数
zp: 基準位置 (offset=0の時のnを示す)
ori_n: n番目の配置候補におけるルーラーの向き。
0のとき水平ルーラー。1のとき垂直ルーラー。
posx_n, posy_n: n番目の配置候補の座標

例

(10, 6, (0, 115, 338), (0, 115, 356), (0, 115, 374), (0, 115, 392), (0, 115, 412), (0, 115, 432), (0, 115, 450), (0, 115, 468), (0, 115, 486), (1, 200, 486))

このとき、offset=0 でのルーラーは、水平方向でY軸の座標は432。
offset=-3 だったら、水平方向でY軸の座標は374。
offset=4 だったら、垂直方向でX軸の座標は200。

極値の例

zp=1 は、マイナス方向にはデータが一つも無いことを示す。
zp=size は、プラス方向にはデータが一つも無いことを示す。

offset は配置位置オフセット

図 2 配置位置リストのデータセット例

B01 および B02

B01

ホールドされている間、ルーラーをマウスポインタの位置を基準に表示する。
ホールドが解除されると、B02 に移動。

B02

B02_1

ホールドが解除された位置を取得 ◇ hrp(x,y)

B02_2

hrp(x,y) を中心に部分 OCR の実施
配置位置リスト 1 を作成

B02_3

hrp(x,y) を中心に文書構造解析を実施
配置位置リスト 2 を作成

B02_4

hrp(xy) における配置位置リスト 1 と 2 を比較。

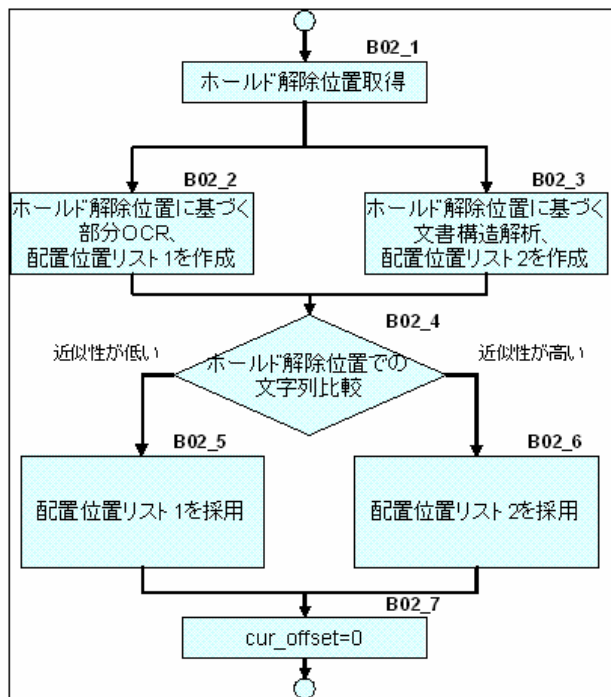


図 3 B02 フローチャート図

近似性が高い場合は B02_5 へ、近似性が低い場合は B02_6 へ。

B02_5

配置位置リスト 1 を採用。

B02_6

配置位置リスト 2 を採用。

B02_7

cur_offset は zero にもどす。

C01, C02 および C03

C01

移動指示分移動した時、現在の配置位置リストの中から候補が得られるかどうかは式 C01 を実行して判断する。

C02

C02_1

移動上限にある配置候補は、 $cur_offset + cur_zp$ が 0 以下の場合 offset が 0 の配置候補。
 $cu_size + 1$ 以上の場合 offset が cur_size の配置候補。

C02_2

(max_posx, max_posy) を中心に部分 OCR の実施

配置位置リスト 1 を作成

C02_3

(max_posx, max_posy) を中心に文書構造解析を実施、配置位置リスト 2 を作成

C02_4

(max_posx, max_posy) における配置位置リスト 1 と 2 を比較。近似性が高い場合は C02_5 へ、近似性が低い場合は C02_6 へ。

C02_5

配置位置リスト 1 を採用。

C02_6

配置位置リスト 2 を採用。

C02_7

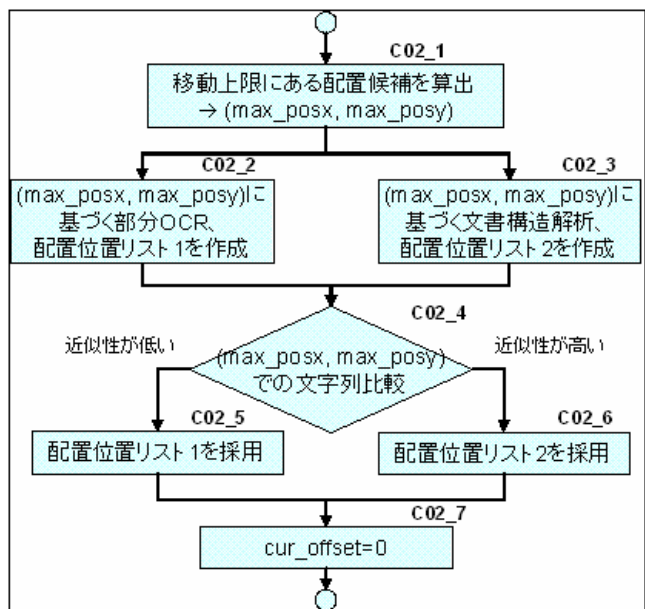


図 4 C02 フローチャート図

式C01

移動指示のあつた距離 → keycount (整数値)
現在の配置位置オフセット → cur_offset
現在の配置位置リストの配置候補データ数 → cur_size
現在の配置位置リストの基準位置 → cur_zp

配置位置リストに配置位置候補がある条件
 $0 < cur_offset + cur_zp + keycount < cur_size + 1$

図 5 式 C01

cur_offset は zero にもどす。

C03

cur_offset := cur_offset + keycount

B02_2 及び C02_2 で行われている部分 OCR の説明

Step 1.

OCR する範囲は、通常は、例のように範囲を限定して行う。常に全体を OCR の対象としても良いが、スクロールが発生した場合や、ページ中身が書き換えられた場合には再度 OCR する必要があるため、システムとの処理速度との兼ね合いでサイズを決める。あるいは利用者の指定でサイズを決めても良い。

ホールド解除位置は、利用者が、ここにルーラーを引きたいという考えが示されているので、これを評価の基準点にする。

Step 2.

青枠は文字列のある範囲、黄色枠は画像のある範囲と認識されたことを示している。青枠(1)については、OCR によって、横書きであること、6 行あることが最低限判別される。OCR によっては、OCR 領域を自動的に拡大するなどして、7 行あることを見つけることもある。(右の例)ルーラー対象エリアは、ホールド解除位置に近いテキストブロック、例の場合は(1) を対象とする。さらに、横書きであるので、横方向ではみ出る部分(2)(3)(4) の OCR 結果は破棄し、(5) (6) のみ残す。(1)と(6)、(5)と(6)の間もルーラーを配置すべき箇所であるが、これを発見できるかどうかは利用する OCR による。

Step 3.

配置位置リストを作成する。(例の場合、X 座標はホールド解除位置に従う)

(9, 3, (0,610,13), (0,610,68), (0,610,120), (0,610,135), (0,610,150), (0,610,165), (0,610,180), (0,610,195), (0,610,210))



図 6 OCR 処理の例

B02_3 及び C02_3 で行われている文書構造解析の説明

文書構造解析の例として、ポインタで示された文書が HTML 文書であった場合を説明する。HTML でなくとも、文書構造の解析が可能な場合は、本文書構造解析の対象となる。

Step 1.

文書構造の解析をするには、指示された文書のソースコードを最初取得する。次に、指示された場所が、ソースコード中のどこなのかを特定する。よって、これが出来ない、文書の場合は、この処理は行われない。(部分 OCR のみで処理される。)

Web コンテンツの場合は、HTML に関しては、取得することができる場合がある。

Step 2.

指示された箇所から前後数行を取得する。

(例の場合は、マイナス方向に 5 行、プラス方向に 6 行)

注目すべきなのは、もっともマイナス方向に移動したところに「サポート」があるが、これは、左ナビゲーションに位置し、むしろ、下方向に移動している。コンテンツの流れにそって適切に候補が得られている。

Step 3.

それぞれのテキストが表示されている位置を取得し、その結果から配置位置リストを作成する。(座標は行の先頭に設定する。)

- (12, 6, (0,20,300),(0,140,13),(0,140,38),(0,140,68),(0,140,103)
- (0,140,120),(0,140,135),(0,140,150),(0,140,165),(0,140,180),(0,140,195),(0,140,210))

HTMLソースを取得、ポイント箇所を特定。

```
<td valign="top"><a href="solution_offerings/EasyWebBrowsing.html#main" class="fbox">Easy Web Browsing&#6482;をよりよく使やすく、機能を強化しました。(2008年5月26日)</a><br /></td><td valign="top"><span class="ipt"></span></td>
```

Step1

Step2 – 表示位置の計算対象となるテキスト (行単位)

```
サポート
アクセシビリティ・センター
リードストーリー
仮想世界におけるアクセシビリティ(2008年7月16日)
アクセシビリティ ニュース
Easy Web Browsing™ をよりよく使やすく、機能を強化しました。(2008年5月26日)
2007 IBM IT キャンプが実施されました。(2007年8月22日)
Easy Web BrowsingがWindows VistaおよびInternet Explorer 7へ対応できるようになりました(2007年7月23日)
字帯編集システムの紹介ページができました。(2007年6月15日)
革新的な科学およびテクノロジーの促進とアクセシブルな情報環境の確立(2007年3月6日)
ホームページリーダーのWindows VistaおよびInternet Explorer 7への対応について(2007年1月30日)
+ ニュース
```

Step3 – 配置位置リストを作成

図 7 文書構造解析例

A02 で行われているルーラーの配置処理の説明(1/3)

本アイデアでは、ルーラーを引く対象とした文字列の、行間隔、文字の大きさ、文字色、背景色、配置位置が分かっているので、それに応じたルーラーを引くことが可能である。以下、可能になることをケースに分けて説明する。

行間の広さに合わせてルーラーの太さを変更

行間が広い場合

ウェブサイト評価サービス
ウェブサイト、ウェブアプリケーションおよびポータル・サイトをアクセシビリティの観点から評価します。

行間が狭い場合

ウェブサイト評価サービス
ウェブサイト、ウェブアプリケーションおよびポータル・サイトをアクセシビリティの観点から評価します。

行間が狭いのに太いルーラーを引いて文字と重なっている例

本アイデアの場合

ウェブサイト評価サービス
ウェブサイト、ウェブアプリケーションおよびポータル・サイトをアクセシビリティの観点から評価します。

行間隔に合わせてルーラー幅を自動変更して配置している例
ルーラー幅は、行間隔の80%等に設定する。最小幅などは、利用者が決めるので良い。

背景色や文字色に配慮してルーラー色を変更

ウェブサイト評価サービス
ウェブサイト、ウェブアプリケーションおよびポータル・サイトをアクセシビリティの観点から評価します。

背景色とルーラー色が近いため、ルーラーが見えにくくなっている。

ウェブサイト評価サービス
ウェブサイト、ウェブアプリケーションおよびポータル・サイトをアクセシビリティの観点から評価します。

背景色に応じてルーラー色を変えている。

ルーラーの色の決め方

- ・背景色の補色にする
- ・背景色だけでなく、文字色にも配慮してルーラーの色を決定する。

図 8 太さ変更および配色変更の例

A02 で行われているルーラーの配置処理の説明(2/3)

ルーラーを階段状に表示することで、段組されているコンテンツであっても、ルーラーを常に文字に重ならないように配置する例

配慮せずにルーラーを配置した例

アクセシビリティ・センター

アクセシビリティ・センター	リードストーリー	EWB 2.0 (新型パネル) 拡大 読み上げ → 新型パネルを使って、ご覧のサイトで文字拡大・読み上げ機能をお試しいただけます
ニュース	仮想世界におけるアクセシビリティ (2008年7月16日)	
ソリューション		
お客様事例・研究事例	アクセシビリティ ニュース	
アクセシビリティ製品情報	→ Easy Web Browsing™ をよりよく使いやすく、機能を強化しました。(2008年5月26日)	
開発者向けガイドライン	→ 2007 IBM IT キャンプが実施されました。(2007年8月22日)	
アクセシビリティ教育	→ Easy Web BrowsingがWindows VistaおよびInternet Explorer 7へ対応できるようになりました(2007年7月23日)	
ACパートナー	→ 字幕編集システムの紹介ページができました。(2007年6月15日)	

配置位置リストを活用してルーラーを配置した例

アクセシビリティ・センター

アクセシビリティ・センター	リードストーリー	EWB 2.0 (新型パネル) 拡大 読み上げ → 新型パネルを使って、ご覧のサイトで文字拡大・読み上げ機能をお試しいただけます
ニュース	仮想世界におけるアクセシビリティ (2008年7月16日)	
ソリューション		
お客様事例・研究事例	アクセシビリティ ニュース	
アクセシビリティ製品情報	→ Easy Web Browsing™ をよりよく使いやすく、機能を強化しました。(2008年5月26日)	
開発者向けガイドライン	→ 2007 IBM IT キャンプが実施されました。(2007年8月22日)	
アクセシビリティ教育	→ Easy Web BrowsingがWindows VistaおよびInternet Explorer 7へ対応できるようになりました(2007年7月23日)	
ACパートナー	→ 字幕編集システムの紹介ページができました。(2007年6月15日)	

図 9 階段状ルーラーの例

A02 で行われているルーラーの配置処理の説明(3/3)

読み上げ応用する例

アンダーラインが引かれた行には、どのような文字が解析済みであるので、読み上げることも可能である。

アクセシビリティ・センター

- ニュース
- ソリューション
- お客様事例・研究事例
- アクセシビリティ製品情報
- 開発者向けガイドライン
- アクセシビリティ教育
- ACパートナー

リードストーリー
仮想世界におけるアクセシビリティ(2008年7月16日)

読み上げ
「仮想世界における・・・」

EWB 2.0 (新型パネル)
拡大
読み上げ
→ 新型パネルを使って、ご覧のサイトで文字拡大・読み上げ機能をお試しいただけます

マスキングの例

ルーラーと同様の目的を持ったマスキング(目的の行以外を隠す機能)やハイライト(目的の行を強調すること)にも利用可能である。

アクセシビリティ・センター

- ニュース
- ソリューション
- お客様事例・研究事例
- アクセシビリティ製品情報
- 開発者向けガイドライン
- アクセシビリティ教育
- ACパートナー

リードストーリー
仮想世界におけるアクセシビリティ(2008年7月16日)

ハイライトの例

色を変えることもできる

EWB 2.0 (新型パネル)
拡大
読み上げ
→ 新型パネルを使って、ご覧のサイトで文字拡大・読み上げ機能をお試しいただけます

図 10 読み上げやマスキングの例