

# 文章のシームレスな表示に関する研究

## —電子テキストの拡大・縮小表示の構想と概要—

杉山正治, 柴田みゆき\*, 生田敦司\*, 齋藤晋\*, 宮下晴輝\*

大谷大学文学部人文情報学科\*

**概要:** 電子テキストはその利便性ゆえに広く受け入れられているが、紙媒体は電子テキストと異なる利便性を保持している。また、ブラウザを使用して電子テキストを表示する場合、ウィンドウサイズによる制約やページ遷移の問題がある。電子テキストの利便性を生かしつつ、書籍の利便性を持たせる工夫として、本研究ではページ遷移が発生しない文章表示システムのプロトタイプを Ajax を用いて構築し、「Magnifying And Simplifying System for Text EXTension Version 1.00 (略称, MaSSTExt V1.00)」と名づけた。本システムは目次や本文などのシームレスな表示/非表示, 及び拡大縮小を実現した。これにより、長大な文章であっても1画面のうちに情報把握が可能になる。

**keyword:** 目次, ページ遷移, 書籍, 表紙

## A Study for Displaying Book with Seamless Pages

—Basic Concept for Magnifying and Simplifying System for Enhanced View of Text—

Seiji Sugiyama, Miyuki Shibata\*, Atsushi Ikuta\*, Susumu Saito\*, Seiki Miyashita\*  
Department of Humane Informatics, Otani University\*

**Abstract:** Despite many conveniences of electronic media, paper media still have advantages because of different conveniences. When using web browser for displaying text files, its disadvantages are clear. We assumed these disadvantages come from limitation of displayable area of window or transition of pages. In this paper, a solution for these disadvantages with our prototype system is introduced: "Magnifying And Simplifying System for Text EXTension Version 1.00 (MaSSTExt V1.00)". This system is made by Ajax and work with web browser. It is enable displaying huge size of text file only one window or seamless display/not-display from index to body or annotation data.

## 1 はじめに

### 1.1 本研究の目的

人間の知識を外部蓄積機器に記憶し、情報交換を容易にするメディアとして、紙媒体は古来より現在に至るまで文化の中心にある。しかし、紙媒体は管理の困難さや保管のための巨大な蓄積場所の確保など問題が多い。また、検索に手間がかかる。

これらを克服する有力な後継メディアとしてコンピュータを利用する電子媒体が登場した。電子媒体は、膨大な情報量をコンパクトに管理保存でき、検索も簡便である。インターネットの発達により、通信を行いながら情報を表示できる環境も整った。これにより、様々な種類のデータがインターネット上で電子媒体として管理・提供されるに至った。

しかし、テキストデータ分野においては、現状ではなお紙媒体による提供が優位である。これには様々な理由が考えられる。従来の Web ブラウザで

は全体の俯瞰的な表示と部分の詳細な表示とを両立させたテキストの可視表現が困難である。この点は、リンク機能によって全体と部分を切り替える方法の採用により回避されているかに見える。ところが、シームレスではないためにページ遷移が発生し、一瞥して関係を把握することが難しい。また、ページ遷移により通信時間のロスが発生する。

そこで、本研究では特に通信を前提とした Web ブラウザによるテキスト表示の問題に着目した。この可視表現の問題とページ遷移の問題を同時に解決する一手法を、制作したプロトタイプシステム「Magnifying And Simplifying System for Text EXTension Version 1.00 (以下, MaSSTExt V1.00 と略す)」を用いて提示する。MaSSTExt V1.00 の有用性は、文書構造の階層毎に表示/非表示処理を行いながら拡大縮小させることにより、電子テキストのシームレスな可視表現を実現することにある。

なお、先行研究事例として、著者らは既にテキストベースの系図表示システムを開発している<sup>[1][2]</sup>。

## 1.2 紙媒体の特徴

紙媒体によるテキストには、付帯情報として紙面の大きさや本の厚みなどの物理量が存在する。多くの場合、1ページの面積や文字数が固定であるページをめくりながら閲覧する。各ページは物理的に連続している。その結果、次のような効果が得られる。

- A1. 全体の分量を体感し把握できる。
- A2. 閲覧したい特定の位置をすばやく開いて詳細に読むことが出来る。
- A3. 前後の文章の見比べも容易である。
- A4. 上記3つの行為が同時に可能である。

## 1.3 電子テキストの特徴と問題点

電子テキストの利点の1つは、メモリ容量の許す限り長大なテキストを1つのファイルに記述できることである。しかし、従来の表示システムは通信速度とウィンドウサイズに依存するため、表示にストレスが生じる。そこで、適切な大ききで分割されたファイルの利用が前提となり、リンクが多数生じる。しかも、一般的にはファイル毎に文章の容量が異なる。したがって、以下の問題が生じる。

- B1. 全体の分量を体感しにくい。
- B2. 閲覧したい特定の位置にアクセスしにくい。
- B3. 前後の見比べは容易ではない。
- B4. 1.2節 A1.~A3.の同時把握は難しい。

これらの問題から、Webブラウザによる電子テキスト表示では、1.2節に示すような紙媒体における利便性を失っていることがわかる。

## 1.4 解決策

読み手に快適なブラウザ表示を行うには、電子テキストにも紙媒体と同様に1.2節 A1.~A4.の特徴が得られればよいと推測できる。

1.2節 A1.を実現するには、テキスト全体を一体化し、1つの空間に収めて全体の俯瞰的な表示を行えばよい。これにより、現在閲覧中の位置も即座に把握できる。

1.2節 A2.を実現するには、文書構造の階層毎に表示/非表示を切り替えるとともに拡大縮小を行い、見出しと本文などをシームレスに表示できればよい。これにより、目次のページやフレームを別途

用意する必要がなくなり、目的の位置を即座に探せる。また、部分の詳細な把握も可能となる。

1.2節 A3.を実現するには、一体化されたテキストを全方位に、シームレスに移動させることができればよい。これにより、1つのテキスト表示においてリンクを使う必要がなくなる。

1.2節 A4.については前述の1.2節 A1.~A3.の解決策が実現できればよい。

これらの要求を満たす類似のシステムとしては、Google Mapなどの地図画像表示システムがある。このシステムでは大きな紙面を持つ情報を表示する際、全体の俯瞰的な把握と、部分の詳細な把握がシームレスに行える。ただし、扱えるのは画像データのみである。

そこで本研究では、電子テキストを表示するシステムに地図画像表示システムのような操作感を取り入れる方法を検討した<sup>†</sup>。

## 1.5 システム構築の基本概念

地図画像表示システムとの比較から、テキスト表示システムにおいて検討すべき要素を提示する。地図画像表示システムでは、全体図と縮尺毎の詳細図があり、それぞれが緯度・経度情報によって関連づけられ、表示位置が決定される。さらに、マウスホイール等の操作により、縮尺が変更され、拡大縮小が実現する。また、マウスドラッグ等の操作により、表示位置が変更される。これらがシームレスに行われる。

テキストにおいて、全体図に相当するのは表題や目次である。詳細図に相当するのは本文などである。テキストに縮尺の概念は存在しないが、その代わりに文書構造の階層毎に異なるフォントサイズが用いられる。テキストにおいて、緯度・経度情報に相当するのは、文書構造の階層毎に与えられるブラウザ上の座標系である。

この読み替えにしたがって、テキストを整形しながら表示し、閲覧者の指示により、表示/非表示の切り替えと拡大縮小を行うことで、シームレスな表示が可能となる。なお、テキストにおいて拡大縮小のイメージを実現するため、フォントサイズと用紙サイズを増減させることにする。

<sup>†</sup>地図画像表示システムのような操作感を取り入れるという表現は、本システムにおいてGoogle Map APIを利用するという意味ではない。XMLに記述されたテキスト情報を表示する時にGoogle Mapのような操作感を得るためのAPIを独自に開発するということである。

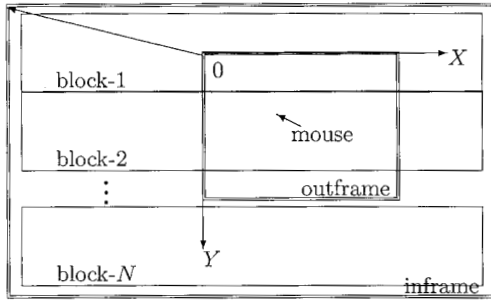


図 1: 座標系の概略図

## 2 MaSSTExt V1.00の概要

### 2.1 使用言語と動作環境

Ajaxは“Asynchronous JavaScript + XML”の略称である<sup>[3][4][5]</sup>。AjaxはJavaScriptを用いてユーザからのアクションに応じ、非同期にHTTP通信を行い、ページ遷移せずに動的に表示内容を変化させる手法であり、シームレスな画面を構築するのに適している。以上より、MaSSTExt V1.00の構築にはAjaxを用いた。なお、動作確認はWindows 2000, xp上のInternet Explorer 6で行った。

### 2.2 座標系の設定

Webブラウザ上での座標系を検討する。図1に示すように表示領域をoutframe、テキストを記述する領域をinframeとする。outframeはブラウザ表示領域に固定されている。inframeの位置はoutframe左上から測った座標として与えられる。inframeには文書構造の階層毎に分割されたN個のblock要素が間隙なく相対座標として並ぶ。outframeより外側のinframeの内容は非表示にしておく。mouse位置とblock位置の比較にはブラウザ表示領域の絶対座標を用いる。

### 2.3 XMLデータフォーマット

MaSSTExt V1.00へのテキストの入力にはXML形式を用いる。書き手への利便性を考慮し、XMLに従来のHTML形式<sup>[6][7]</sup>をほぼそのまま流用する。すなわち、図2に示すように、XMLのルートタグを<html>とし、<head>内部に初期設定を、<body>内部に本文を、それぞれ記述する。

```
<?xml version="1.0" encoding="utf-8"?>
<html>
  <head>
    <width>inframe 基準幅  $w_0$ </width>
  </head>
  <body>
    <h1>文書のタイトル</h1>      …レイヤ0
    <h2>章のタイトル</h2>       …レイヤ1
    <h3>節のタイトル</h3>       …レイヤ2
    <h4>項のタイトル</h4>       …レイヤ3
    <p>
      段落 (文章)                …レイヤ4
    <q>図枠</q>                  …レイヤ5
    <cite>脚注等</cite>          …レイヤ6
    </p>
    <table>数式 (表) </table>    …レイヤ4
    <dl>箇条書き</dl>           …レイヤ4
  </body>
</html>
```

図 2: XML タグ一覧

responseXMLを用いてXMLデータを読み込んだ後、再帰処理によってタグの入れ子と属性を取得する。これにより、一部制限<sup>‡</sup>はあるが<body>内部に殆どのHTMLタグをそのまま記述できる。

このうち、図2の<body>内部に示したタグには文書構造の階層を表すレイヤ番号が付与される。図枠と脚注等については適切なHTMLタグが無いので、便宜上<q>と<cite>を用いて、内部的に本システム特有の表示を行う(2.8節参照)。なお、初期設定として<width>を記述すればinframeの基準幅を指定出来る(2.5節参照)。

### 2.4 フォントサイズ曲線

MaSSTExt V1.00ではフォントサイズを変更することにより画面の拡大縮小を実現する。この操作は、マウスホイールの増減をカウントしながら行われる。フォントサイズには相対単位である[em]<sup>§</sup>を用いる。これにより、ブラウザの文字のサイズを変更するメニューを利用でき、閲覧者の環境に合わせた基準の文字サイズを選択できる。

フォントサイズが変更されても、各レイヤ間のフォントサイズの倍率は維持しなければならない。これは文書構造の階層の区切れを明確にするためである。また、閲覧者に拡大縮小していることを伝えるため、常に全てのレイヤのフォントサイズが変化することが望ましい。表示/非表示の移行時には特定のレイヤについてのみ、少ない回数で拡大縮小が実行されるようにしたい。

<sup>‡</sup>XHTML準拠、かつ、本システム特有のタグ設定がある。

<sup>§</sup>その要素のフォントサイズの値を1.0とする。

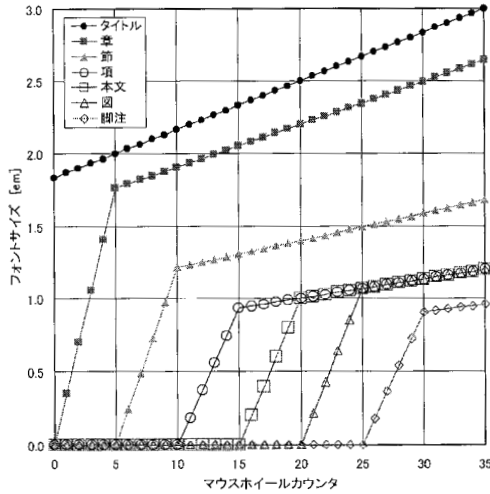


図 3: フォントサイズ曲線

これらの要求を実現するためのフォントサイズ曲線を定義する。レイヤ番号を  $l$  とし、 $l=0$  の倍率を 1.0 とした時のレイヤ間倍率  $M(l)$  を式(1)のように与えておく。

$$M(l) = \left( \frac{25}{25}, \frac{22}{25}, \frac{14}{25}, \frac{10}{25}, \frac{10}{25}, \frac{10}{25}, \frac{8}{25} \right) \quad (1)$$

常に表示されるトップレイヤには一次関数を用いる。この直線の傾きを  $d = 1/30$ 、最大レイヤ番号を  $l_{max}$ 、表示/非表示の移行回数を  $N_r$  とし、マウスホイール最大カウントを  $C = l_{max}N_r$  とすれば、式(2a)~(3c)によりフォントサイズ曲線  $F(l, n)$  が決定される<sup>¶</sup>。この曲線のグラフを図3に示す。

$l=0$  のとき

$$F(0, 0) = 1.83333 \quad (2a)$$

$$F(0, n) = F(0, 0) + dn; (1 \leq n \leq C) \quad (2b)$$

$l \geq 1$  のとき

$$F(l, n) = F(0, n)M(l); (N_r l \leq n \leq C) \quad (3a)$$

$$F(l, n) = F(l, N_r l) \frac{N_r - (N_r l - n)}{N_r}; \\ (N_r l - N_r \leq n \leq N_r l - 1) \quad (3b)$$

$$F(l, n) = 0; (0 \leq n \leq N_r l - N_r - 1) \quad (3c)$$

## 2.5 inframe の幅

inframe の幅は基準幅  $w_0$  ([em] 単位) を元に計算される。フォントサイズが拡大された時に inframe

<sup>¶</sup>本文レイヤ ( $l=4$ ) の非表示への移行直前のフォントサイズが 1.0[em] になるように初期値  $F(0, 0)$  を計算した。

の幅が  $w_0$  に固定されていると 1 行の文字数が徐々に減ってくる。また、フォントサイズのみ拡大すると用紙が拡大された印象が薄くなる。そこで、マウスホイールのカウント  $n$  を用いて変更後の inframe 幅  $w_{ne}$  を式(4)のように計算する<sup>||</sup>。

$$w_{ne} = w_0 + \frac{n}{2} \quad (4)$$

なお、図の拡大縮小については inframe の幅を 100% とし、表示/非表示の移行時に、図の幅を 0% から 100% まで変化させる方式とした。

## 2.6 拡大縮小に伴う座標変更

拡大縮小を行う時には常にマウス位置が拡大縮小の中心になるようにした方がよい。画像データであればこれを実現するのは簡単である。変更前と変更後それぞれの画像のサイズを求め、マウス位置との比が同じになるように位置を決定すればよい。

ところが、表示/非表示、拡大縮小を伴うテキストデータの場合には、単純に inframe の変更前と変更後の位置とサイズを求めるだけでは上手く行かない。これを解決するにはマウス位置の block 要素を特定し、これに画像と同じ手法を用いる。さらに変更後のマウス位置の block 要素と inframe の位置関係を求め、変更後の位置を確定する。

マウス座標を  $(x_m, y_m)$ 、変更前のマウス位置の block 要素左上座標を  $(x_{oe}, y_{oe})$ 、幅を  $w_{oe}$ 、高さを  $h_{oe}$ 、変更後のマウス位置の block 要素左上座標を  $(x_{ne}, y_{ne})$ 、幅を  $w_{ne}$ 、高さを  $h_{ne}$ 、outframe 左上座標を  $(x_o, y_o)$ 、変更後の inframe 左上座標を  $(x_i, y_i)$  とすれば変更倍率  $(r_x, r_y)$  は式(5)で与えられる。

$$r_x = \frac{|x_m - x_{oe}|}{w_{oe}}, \quad r_y = \frac{|y_m - y_{oe}|}{h_{oe}} \quad (5)$$

変更後のマウス位置の block 要素と inframe の差分  $(x_d, y_d)$  は式(6)で与えられる。

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} x_{ne} \\ y_{ne} \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (6)$$

最終的に変更後の inframe の位置  $(x_{ni}, y_{ni})$  は式(7)で与えられる\*\*。

$$\begin{bmatrix} x_{ni} \\ y_{ni} \end{bmatrix} = \begin{bmatrix} x_m \\ y_m \end{bmatrix} - \begin{bmatrix} w_{ne} r_x \\ h_{ne} r_y \end{bmatrix} - \begin{bmatrix} x_d \\ y_d \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \end{bmatrix} \quad (7)$$

<sup>||</sup>基準幅  $w_0$  については XML 読込後に値を変更できるので、閲覧者の環境に inframe の基準幅を合わせることができる。

\*\*厳密には要素の枠の太さ等を考慮して計算する必要があるが、それらの補正式については省略する。



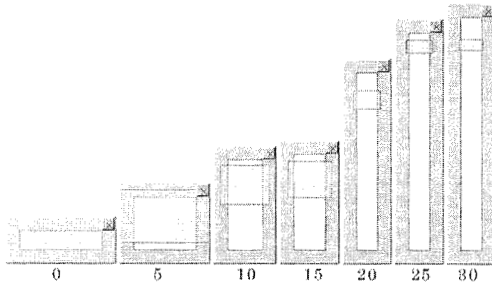


図 4: インジケータの拡大縮小の様子

なお、変更前のマウス位置の block 要素が非表示となり消滅している場合には、消滅していない直上の block 要素を探査し、これを変更後の位置とサイズと見なす。さらにマウス位置がジャンプしないようにするため、この要素の bottom にマウスがあると仮定し、 $r_y = 1$  において式 (7) の  $y_{ni}$  を計算する。

## 2.7 インジケータの大きさ

MaSSTExt V1.00 では地図画像表示システムと同じく、テキストを全方位にシームレスに移動できる。したがって全体を示すインジケータにも全方位に移動可能なものが必要である。地図画像表示システムのインジケータでは一回り大きな領域しか表示されていない。これに対し、テキスト表示では常に全体が見えている必要がある。これはスクロールバーと同じ考え方である。

そこで、全方位に移動可能なスクロールバーを実現するため、インジケータ自体を伸縮できるようにした。現在の inframe の縦横比と大きさによってその形を決定する。インジケータに表示される inframe の幅  $w_{ii}$ 、高さ  $h_{ii}$  には式 (8) を用いた。

$$w_{ii} = \log \left( \frac{w_i}{h_i} + 1 \right) w_{i0}, \quad h_{ii} = \log \left( \frac{h_i}{w_i} + 1 \right) h_{i0} \quad (8)$$

ここで、 $w_{i0}$  は基準幅、 $h_{i0}$  は基準高さを表す。なお、 $w_{ii}$ 、 $h_{ii}$  の値は outframe の大きさとバランスを取る必要がある。今回は式 (9) の制限を設けた。

$$25 < w_{ii} < w_{i0}, \quad h_{ii} < \frac{h_o}{2} \quad (9)$$

図 4 にマウスホイールカウンタによってインジケータが変化の様子を示す。なお、inframe の高さは現在表示設定になっている全 block 要素の高さの合計であり、テキストの分量によってインジケータの高さは不連続に変化する。

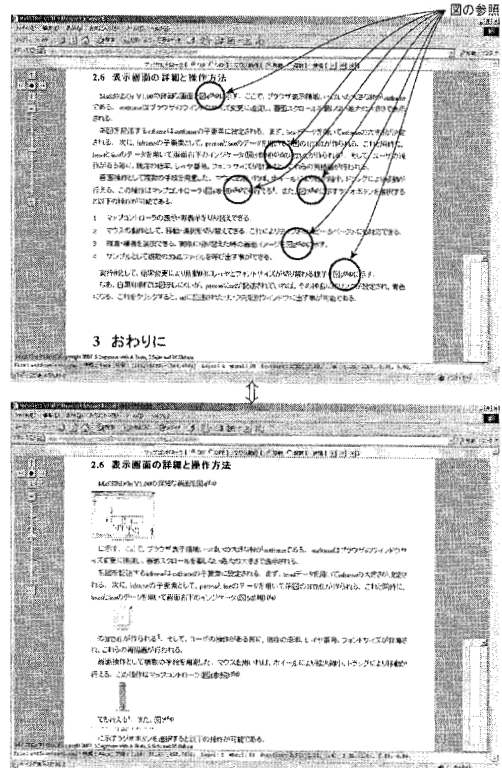


図 5: 図の表示／非表示の様子

## 2.8 図・脚注等の表示

図や脚注は本文を読んでいる時には特に見えていなくても問題ない。閲覧者が図や脚注を見たい時にのみ表示させればよい。また、ブラウザ上では紙面の大きさに制限は無い。そこで、MaSSTExt V1.00 では、図や脚注は文章と文章の間に空間を作って表示することにした。

この方式では、説明を要する本文の直下に図や脚注を配置出来るので、極めて少ない画面スクロールでシームレスに閲覧可能である。また、図や脚注が長くなっても問題なく表示出来る。

図の表示／非表示の様子を図 5 に示す。文章中に図の参照として (fig) マークを表示し、図があることを閲覧者に知らせる。次に、マウスホイールを用いて拡大すると、(fig) マークを境にして本文が 2 つに分断され、その間に図が現れる。ここでは例示しないが、さらに拡大すると図の幅は inframe いっぱいまで大きくなる。図を見る必要が無ければ縮小することにより、文章のみに戻すことができる。

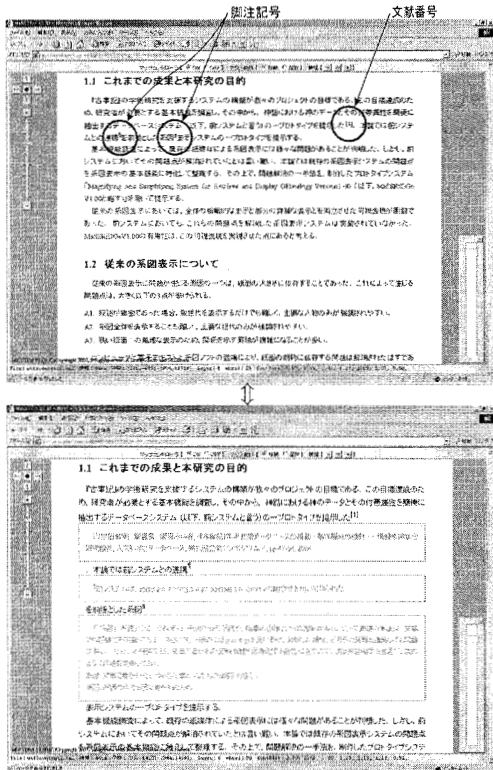


図 6: 脚注の表示／非表示の様子

同様に、脚注の表示／非表示の様子を図6に示す。脚注も図と同じ方法で表示／非表示が変更される。脚注は、一般の注釈以外に、参考文献等の引用情報の表示に利用してもよい。脚注は細切れに複数設定しても、それぞれが独立した空間を作って表示される。図のレイヤ番号は5、脚注のレイヤ番号は6としているので、図の幅が100%になった後に脚注が表示されるようになる点だけが異なる。

さらに、この考え方を応用すれば<CITE>を用いるだけで訳文（英語など）の記述も行える。訳文の表示／非表示の様子を図7に示す。

## 2.9 操作方法の詳細

表示／非表示の切り替え処理は文書構造全体にわたって同一のアルゴリズムが適用される。すなわち、全ての階層の切り替えはマウスホイールの増減のみで行える。マウスホイールにより自動的に階層構造とフォントサイズが切り替わる様子を図8に示す。

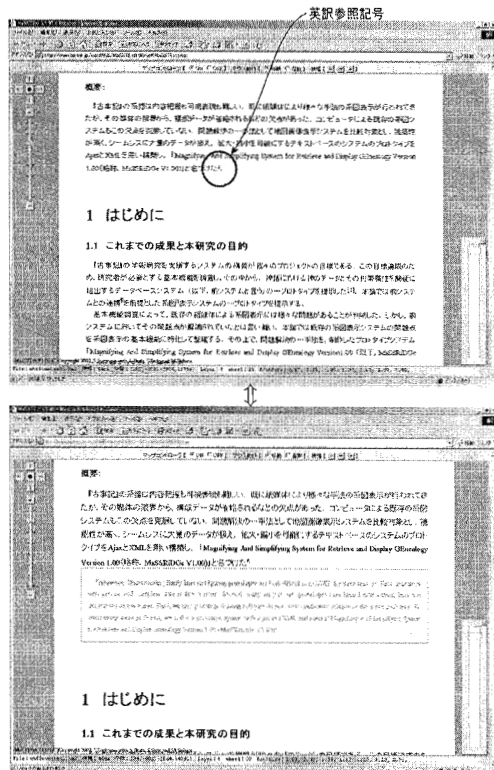


図 7: 訳文の表示／非表示の様子

なお、マウスホイールの代わりにoutframe左上に配置したマップコントローラを用いることもできる。マップコントローラの横線はレイヤの切り替わるポイントを表す。文書構造の階層を一気にジャンプして最上層、最下層レイヤに移動することもできる。また、即座に特定のレイヤに移動することも可能である。

画面上部にはマウスの動作として、移動モードと選択モードの切り替えボタンが用意されている。選択モードに切り替えれば表示中のテキストを選択しコピー・ペーストすることができる。この機能は、ブラウザに通常のHTMLとしてテキストを表示しているの、Windows標準のコピー操作を行えるように設定を切り替えるだけで実現できる。

この他、inframeの基準幅を変更するボタンが用意されている。ウインドウサイズが小さい場合などにはこれを用いてテキストの幅を適宜縮小できる。また、inframe幅を大きくすれば、それにとまって図枠も大きくなる。これにより図などを大きく見たい場合にも対応可能である。



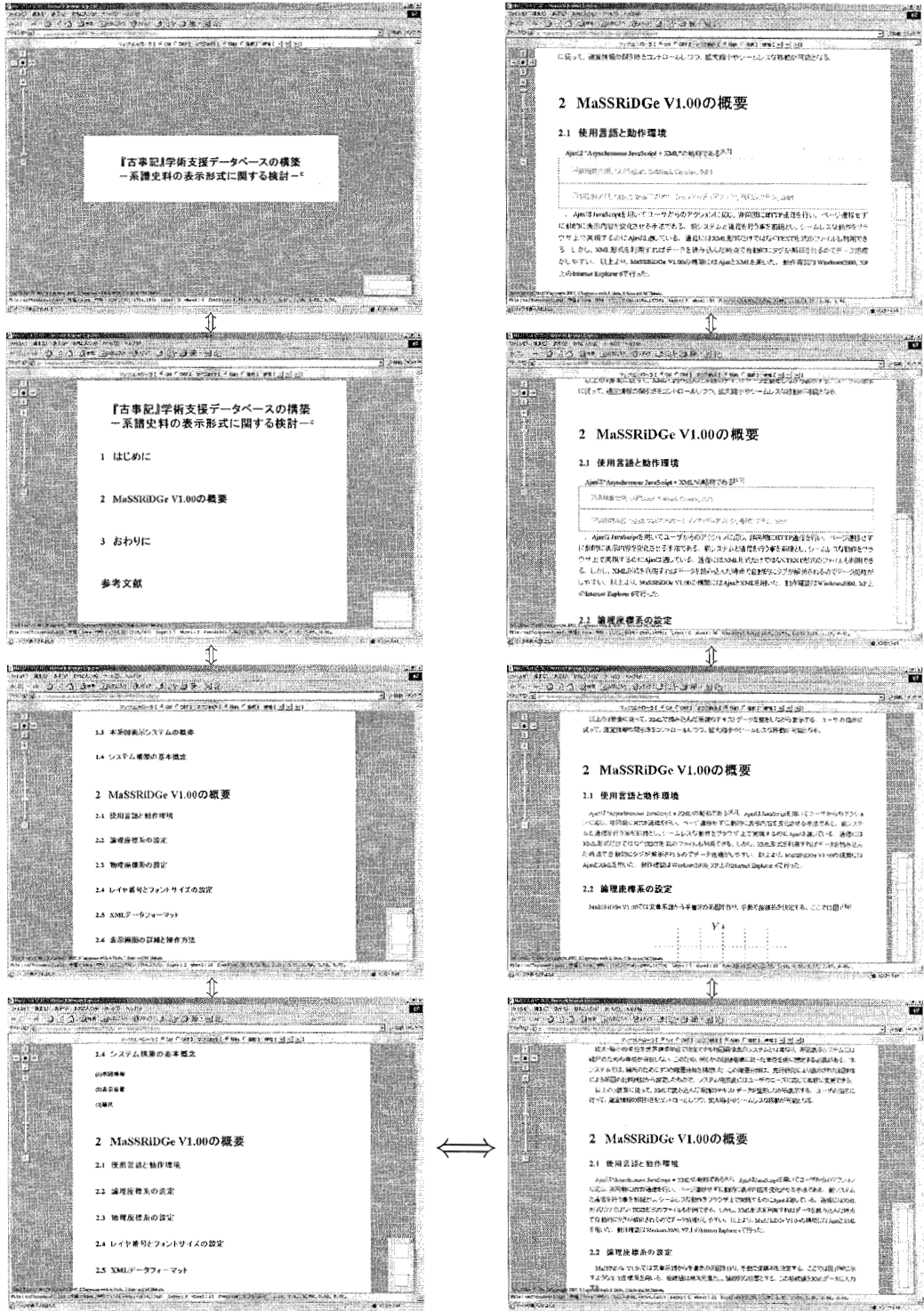


図 8: マウスホイールにより自動的に階層構造とフォントサイズが切り替わる様子

## 3 システムの検証

### 3.1 ブラウザによるHTML表示

テキストには章・節・項・段落などの文書構造がある。HTMLは本来この文書構造を規定するための書式であり、そのためのフォーマットがあるので書き手への利便性は確保されている。また、このフォーマットが厳格に守られていれば、検索等への応用に関する利便性も確保されている。

ところが、これまではHTMLで厳格な文書構造を書いたとしても、ブラウザは自動的に目次を生成したり、テキストの全体像を見せたりすることはしなかった。その結果、全体の俯瞰的な把握と部分の詳細な把握を同時に可能にする表示方法については、書き手の裁量に任される結果となっていた。

しかし、そのような煩雑な作業を好んで行う書き手は少ない。その上、特にタグを用いなくてもテキストは画面上に表示されてしまうため、文書構造は無視され、書き手はワープロのように自由なフォントサイズ指定や自由なレイアウトを好むようになった。

これに対し、MaSSTExt V1.00では、厳格な文書構造を用いてテキストを記述するだけでシームレスな表示が可能であり、このようなHTML記述の曖昧さを解決できる。

### 3.2 リンク構造

通信の問題とウインドウサイズの問題から、多くの場合、1つのテキストが2つ以上のファイルに分割され、リンクにより連結される構造が用いられる。このような構造の中でも全体の把握を行うために、目次の画面が別途用意され、見出しにリンクが設定されることが多い。

電子テキストの1画面の長さは可変であるため、全体の容量がどの程度であるかは予測できない場合が殆どである。また、リンク先にどの程度の容量のテキストがあるかという情報は一般的には表示されない。フレームを用いて目次と本文を同時に表示させる方法もあるが、物理的に分断された可変長のテキストが表示される状態は変わらない。

連続したテキストを見比べて前後関係を把握する行為をWebブラウザで試みると、何度もリンクボタンを押すことになる。非同期通信やキャッシュを用いて高速にはできるが、読み手にとっては煩雑

な動作である。

この問題を解決するため、1つのテキストをできるだけ1つのファイルに収めて表示することも行われる。内容が長大な場合には全体の俯瞰的な把握が困難になるので、目次を配置し、ページ内リンクを用いることがよく行われる。ところが、本文へジャンプすると今度は目次が見えなくなり、迷子になりやすい。したがって文章の所々に「戻る」ボタンや目次が何度もみられる場合がある。

結局、既存の表示方法では複雑なリンク構造が多用され、全体の俯瞰的な把握と部分の詳細な把握の両立は困難である。

これに対し、MaSSTExt V1.00では、1つのテキストに関してはリンクを設定する必要がなくなるので、このような問題が全て解消される。

## 4 おわりに

本研究では、電子テキストの表示システムとして既存の地図画像表示システムのような操作感を実現するシステムの基本機能を満足するプロトタイプを制作した。電子テキストの表示システムとしては他に検討すべき課題が多い。それらの課題を解決したシステムを今後も検討、改良していく予定である。

## 参考文献

- [1] 杉山正治, 齋藤晋, 生田敦司, 柴田みゆき, “『古事記』学術支援データベースの構築—系譜史料の表示形式に関する検討—”, 情報処理学会, 第75回人文科学とコンピュータ, 2007-CH-75(7), pp.47-54, 2007
- [2] 柴田みゆき, 杉山正治, 生田敦司, 齋藤晋, 宮下晴輝, “『古事記』学術支援データベースの構築—神話系譜史料の表示形式に関する検討—”, 情報処理学会, 第76回人文科学とコンピュータ, 2007-CH-76(9), 2007
- [3] 高橋登史朗, “入門 Ajax”, SoftBank Creative, 2005
- [4] 羽田野太巳, “AJAX Web アプリケーションアイデアブック”, 秀和システム, 2005
- [5] 川俣晶, “実例で学ぶ [入門と実践] Ajax + XML”, 技術評論社, 2006
- [6] ノマド・ワークス, “詳細HTML基本活用辞典”, 新星出版社, 2000
- [7] アンク, “ホームページ辞典第3版”, 翔泳社, 2002