

## Web ブラウザを利用した文書内挿機能の実現

三浦 元喜<sup>†</sup> 志築 文太郎<sup>†</sup> 田中 二郎<sup>†</sup>

従来の Web 閲覧は、基本的に 1 つの Web ページを 1 つのウィンドウに表示して行われていた。そのため、リンクをたどった際にリンク元文書が画面から失われ、読者は閲覧の文脈を保存できなかった。我々は、リンク先文書をリンク元文書内に展開表示する「文書内挿」を Web 文書の閲覧に適用するシステム **inlineLink** を開発した。**inlineLink** は、従来の文書内挿における問題を解消するため、文書内の移動にかかる手間を軽減する機構と、文書全体を把握するためのナビゲーション機構を備えている。既存の Web ブラウザ上で文書内挿機能を実現するため、Dynamic HTML 技術を用いて **inlineLink** を実装した。評価実験の結果、文書内挿機能が Web 文書閲覧におけるマウスクリック数と移動量を軽減することを確認した。

### Realization of Inline Expansion Functions on a Conventional Web Browser

MOTOKI MIURA,<sup>†</sup> BUNTAROU SHIZUKI<sup>†</sup> and JIRO TANAKA<sup>†</sup>

Conventional web browsing is performed by displaying a web page inside one window. Following of a link in conventional web browsing replaces the previous document entirely, and the readers tend to lose the context. We have designed a system **inlineLink**, which applies an inline expansion method to web browsing. The inline expansion means to insert the linked document after the link anchor. **inlineLink** provides navigation mechanisms such as automatic animated scrolling, zooming and index-jumping in order to reduce the scrolling tasks while handling the longer inlined documents. We have adopted Dynamic HTML to implement the inline expansion functions. Casual users can try them on conventional web browsers. The results of our experiment prove the advantage of **inlineLink** in click counts and the length of mouse movement.

#### 1. はじめに

一般的な Web ブラウザはリンクアンカーをクリックすることによる文書の移動を基本的な機能として備えている。これらの Web ブラウザにおいて、リンクアンカーを選択しリンク先文書を表示すると、リンク元文書は画面から失われる。リンク元文書には、リンク先文書に関連する情報が表示されていることが多いため、リンク元文書とリンク先文書を同時に表示したい場面は多い。

近年普及している Web ブラウザは、リンク元文書の表示を残すためリンク先文書を新しいウィンドウに表示する機能を備えている。しかし、新しいウィンドウの位置や大きさ、重なり順を調整する操作が必要となることが多く、本来の閲覧作業を妨げやすい。新しいウィンドウを生成しない操作として、リンクアンカーを別のウィンドウにドラッグすることによって文書を表示する操作を可能としている Web ブラウザも存在す

る。しかし複数のリンク先文書をそれぞれ個別のウィンドウに表示した場合、読者はアンカーとウィンドウの対応付けを記憶するのが困難になる。

#### 2. inlineLink

我々は、上に挙げた問題を解決し、効果的な Web 文書の閲覧を行えるようにするため“inlineLink”を開発した<sup>1)</sup>。**inlineLink** は、Web 文書閲覧において文書内挿機能を実現するためのシステムである。

##### 2.1 文書内挿

文書内挿とは、文書の一部を置き替えることによってリンク先文書をリンク元文書内に展開表示することを指す。文書内挿の概念は 1982 年に Peter Brown が開発したハイパーテキストシステム Guide<sup>2)</sup> において Replacement-button として実現されている。文書内に配置された Replacement-button を押すと、関連付けられた文書に置き替わるため、読者はより詳しい情報を参照することができる。

Guide においてはコンテンツ作成者は Guide 用のハイパーテキストを用意する必要があったが、**inlineLink** では Guide における Replacement-button の機構を Web

<sup>†</sup> 筑波大学 電子・情報工学系  
Institute of Information Sciences and Electronics  
University of Tsukuba

文書 (HTML によってタグ付けされた文書) に適用した `inlineLink` では, Web 文書における通常のリンクアンカーの代わりに, 文書内挿を行う特殊なリンクアンカー (文書内挿リンクアンカー) を用いる. 文書内挿リンクアンカーは, リンク先文書を取得してリンク元文書内に展開する Open Anchor と, 展開している文書を閉じる操作を行う Close Anchor から構成される.

図 1 に, `inlineLink` における文書内挿リンクアンカーの動作を示す. 図 1 左の Open Anchor をクリックすると, 図 1 右に示すようにリンク先文書がリンクアンカーの後方に展開 (内挿) される. 同時に, Open Anchor は Close Anchor に置き替わる. リンク先文書の終点にも, Close Anchor が現れる. 展開されたリンク先文書は図 1 右に示すように, 視覚表現 (枠やインデント) によってリンク元文書と区別される.

## 2.2 内挿文書の表現方法

Guide と `inlineLink` では, 内挿文書の表現に違いがある. Guide では, 内挿文書と元の文書との区別を意識させないように表示する. これに対し `inlineLink` では, 「枠で囲う」等の視覚表現によって内挿文書の範囲および階層を明確に示すよう設計している. 我々は, 内挿文書の範囲および階層を明示することによって, 読者がハイパーテキスト内における現在の参照位置を意識しやすくなると考えている. なぜなら, この「枠」は読者がリンクアンカーをたどる操作と対応して発生するため, 履歴およびサイトマップとして機能すると考えられるからである. 履歴およびサイトマップはナビゲーション支援における重要な機能である<sup>3)</sup>.

## 2.3 文書内挿に適した Web 文書の特性

`inlineLink` は, Web 文書閲覧において文書内挿機能を実現するシステムである. しかし, 文書内挿は, その性質上, 全ての Web 文書について有効に機能するとは限らない.

Web 文書が以下の特徴を同時に満たす場合には, 文書内挿は有効性を発揮すると考えられる.

- 内挿される文書が短い場合 (例えば, 注釈や用語集等). 内挿後, リンク元文書の多くが画面に残るため, 読者はリンク先文書とリンク元文書を同時に閲覧できる.
- 複数の文書が密接な関連を持つ場合. 1 つの内容が複数の文書に分離している場合 (例として, メーリングリストのアーカイブや掲示板の記事等). 文書内挿は, リンク元文書とリンク先文書のつながりを自然に表現する. そのため, メーリングリストのアーカイブにおいて, 議論の内容を順に追っていくような場合に有効であると考えられる.

逆に, Web 文書が以下のような場合には, 文書内挿の効果は薄いと考えられる.

- 内挿される文書がウィンドウの高さに対して長い場合. 文書内挿によってリンク元文書が画面から失われるため, 同時に閲覧することが困難になる.
- リンク元文書とリンク先文書の関連が薄い場合. 同時に閲覧する必然性が乏しい場合には, 文書内挿の効果は薄い.

また, Web 上にはフレームやテーブルを用いたレイアウトがなされている文書も存在する. このような文書を内挿した場合, 内挿文書の表示がレイアウトによって制限されるため, 表示が崩れたり読みづらくなるため, 内挿の効果を得られにくい. また, 文書制作者の意図しない表示になってしまうため, このような文書は文書内挿には不向きである.

## 2.4 新しいウィンドウを開く操作と文書内挿の比較

Web ブラウザには, リンク先文書を新しいウィンドウとして開く機能を備えているものがある. この機能を用いた場合, 文書の表示は従来から行われている閲覧と同様であるため文書内挿と比べて表示の違和感は少ない. また, ウィンドウの大きさや位置, 重なり順を読者が自由に設定できるというメリットがある. しかし, 操作における自由度が大きい分, 読者に与える負担は大きい. ウィンドウ操作のためのマウス操作量が増加することに加え, 読者はウィンドウ間のリンク関係を記憶しておく必要がある.

## 2.5 文書内挿の問題と解決法

文書内挿機能は, 1 章で述べた問題を解決する点で有効である. しかし, 一般的に文書内挿機能を用いた閲覧には以下の問題点がある.

- (1) ウィンドウの高さよりも, 表示に必要な領域の高さが長い文書を内挿すると, リンクアンカーの下にあった文書が画面から失われる.
- (2) 文書内挿を繰り返すと, ウィンドウ内の文書が長大化 (表示に必要な領域が縦方向に長くなること) し移動にかかるスクロール量が増加する. また, 文書全体の構成を把握することが困難になる.

我々は, これらの問題を緩和するため, (1) 部分内挿機能 (2) ナビゲーション支援機能 (長大化した文書における移動および把握を容易にする機能) の 2 つの機能を実装した.

### 2.5.1 部分内挿

部分内挿とは, 図 2 に示すようにリンク先文書の一部のみを画面に表示する機能である. `inlineLink` に

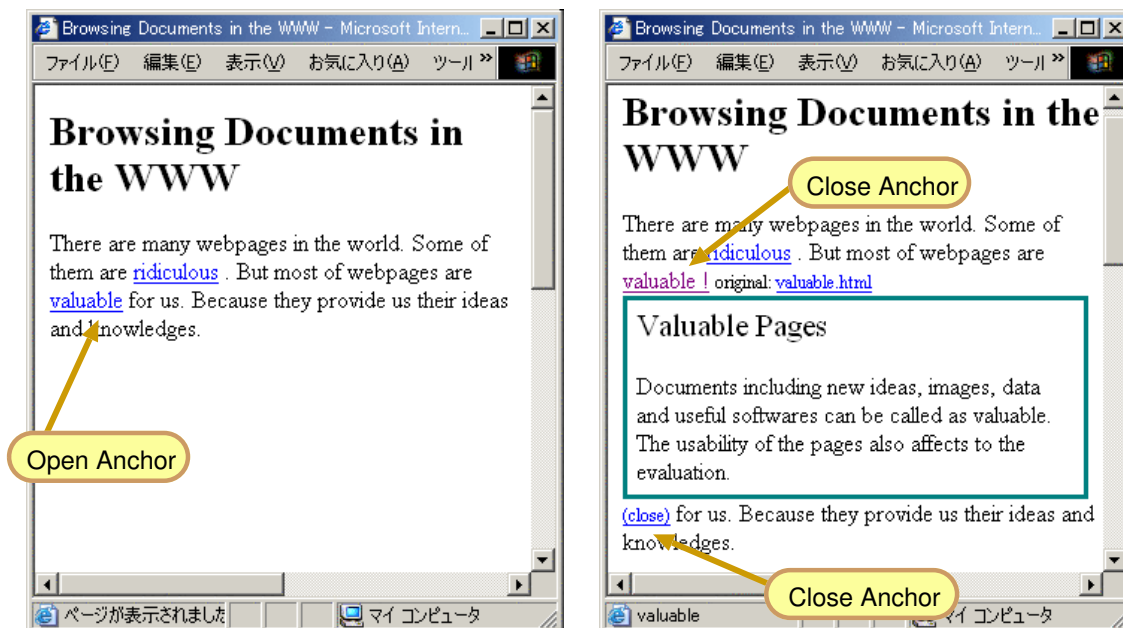


図 1 inlineLink における文書内挿動作

Fig. 1 An example behavior of link anchors in inlineLink

おける部分内挿機能は (1) リンク先文書の表示に必要な領域の高さがウィンドウの高さよりも長い場合 (2) リンクアンカーの下にある文書が重要な場合 (3) リンク先文書の一部分のみを表示したい場合 に効果的である。読者は通常モードと部分内挿モードを切り替えることによって利用する。部分内挿に使われる領域の高さは読者がリサイズ用のアンカー (図 2 における “expand”, “shrink”) をクリックすることにより調整できる。

### 2.5.2 ナビゲーション支援機能

ナビゲーション支援機能とは、長大な文書における読者の移動および現在位置の把握を支援する機能である。inlineLink におけるナビゲーション支援機能は、以下の 3 つである。

- (1) 表示位置を調整するための自動スクロール機能
- (2) 長大な文書内において読者が現在位置を把握するためのズーム機能
- (3) 長大な文書内における移動を効率良く行うためのインデックス機能

#### 自動スクロール機能

読者が Open Anchor をクリックした場合、内挿するリンク先文書に興味を持つと判断し、内挿文書を注目点とみなし画面の中央に表示する機能を提供する。もしウィンドウの高さよりも注目する内挿文書の表示領域の高さが長い場合には、その内挿文書の先頭をウィ

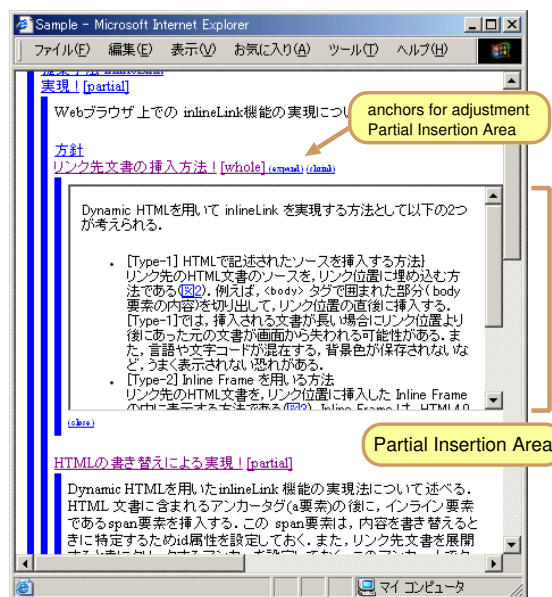


図 2 部分内挿

Fig. 2 Partial insertion

ンドウの上端に揃える。また、読者が画面内の内挿文書をマウスを用いてクリックした場合も注目点として移動を行う。これらの移動にかかるスクロール動作は、アニメーションを用いて急激な画面変化を抑える。自動スクロール機能がない場合、読者は多くの場合文書

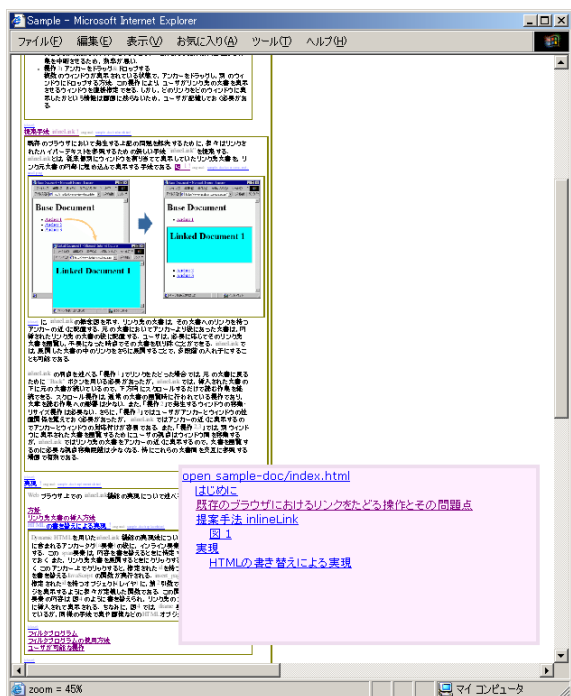


図3 ズーミング機能とインデックス機能：インデックスはマウスカーソル付近に表示される。

Fig. 3 Zooming out and Index List functions: Index List appears near the mouse cursor.

内挿操作後、下方向にスクロールする必要があった。

**inlineLink** では内挿文書を閉じる際にもアニメーションを用いる。Guide を用いた実験報告<sup>4)</sup>によると、内挿文書を閉じる際の急激な画面変化によって読者が混乱することが述べられている。**inlineLink** では、閉じる動作をなめらかなアニメーションによって表現することにより急激な画面変化を抑える。加えて、閉じる内挿文書の開始位置にある文書内挿リンクアンカーを注目点とした自動スクロールを行うことにより、読者のメンタルマップを保つよう設計している。

#### ズーミング機能

通常では一画面に納まらない長大な文書全体を1つの画面に表示するために、ズーミング機能を提供する。縮小操作によって文書全体を表示することにより、読者が文書の概略および現在位置を把握しやすくなると考えられる。拡大および縮小操作は、マウスの横方向のドラッグ操作によって行う(右方向にドラッグすると拡大し、左方向にドラッグすると縮小する)。これは Jazz<sup>5)</sup> における操作に準拠している。図3に、ズーミング表示によって文書を縮小した画面を示す。

#### インデックス機能

内挿を繰り返すことによって、文書が極めて長大とな

り、ズーミング機能を用いても全体を把握するのが困難な場合がある。そのような場合に備えて、**inlineLink** では現在内挿している文書のリンクアンカーを一覧して表示するインデックス機能を提供する。図3に示すように、インデックスの各項目(リンクアンカーの文字列)は階層に対応して表示する。インデックスのアンカーをクリックすると、その文書を注目点とした自動スクロールを行う。インデックスは、読者がキーボードのショートカットから呼び出すことにより、マウスカーソルの近くにポップアップするウィンドウ内に表示される。

また、付加的な機能として、内挿文書内の任意の位置でダブルクリックを行うと文書を閉じる機能を提供する。

### 3. 既存のブラウザによる **inlineLink** の実現

我々は、Web ブラウザ上で **inlineLink** を実現するために、Dynamic HTML と呼ばれる動的な文書変更技術を利用する。HTML 文書の内容は、World Wide Web Consortium (W3C) が標準化した Document Object Model (DOM)<sup>6)</sup> によって階層構造を持つオブジェクトとして扱うことができる。

Dynamic HTML を用いることにより、一般に広く利用されている Web ブラウザ上で動作可能となる。文書内挿リンク機能をブラウザを拡張して実装する場合に比べると自由な動作表現を行いにくい部分もあるが、読者が慣れ親しんだ環境を利用できる点と、手軽に利用できる点を重視した。

#### 3.1 内挿するための仕組み

**inlineLink** における文書内挿は、以下の2つの処理により実現可能である。

- (1) リンク先文書ソースの取得
- (2) 文書ソースの書替え

なお、Close Anchor の動作時には新たに文書ソースを取得する必要がないため「文書ソースの書替え」処理のみを行う。

##### 3.1.1 リンク先文書ソースの取得

任意のリンク先文書ソースを取得するために、**Inline Frame** を用いる。**Inline Frame** は、通常のフレームと異なり、文書中にオブジェクトとして挿入可能なフレームである。**Inline Frame** は HTML4.0 Transitional DTD<sup>7)</sup> において `iframe` 要素として定義されている。文書中において、

```
<iframe src=URI></iframe>
```

と記述すると、任意の URI (Uniform Resource Identifier) で指定された文書ソースが **Inline Frame** に読み込

まれる。この Inline Frame は文書ソース取得が目的であるため、画面に表示されないよう非表示に設定してある。

### 3.1.2 文書ソースの書替え

内挿文書を表示する際に、書替えを行うリンク元文書ソースの範囲を指定するために、以下の2つの処理を行う。(1) 文書ソース内のリンクアンカーをインライン要素 `span` にて囲み、一意な ID を設定する。(2) 文書ソース内のリンクアンカーを文書内挿リンクアンカー (Open Anchor) に変更する。この2つの処理により、図4の “[original anchor]” が “[open anchor]” に変化する。

Open Anchor がクリックされたときに呼び出される JScript 関数 `insert_page()` は、以下の3つの操作を行う。(1) Open Anchor を含む `span` 要素の内容自身を書き替えて、リンク先文書を表示するための `div` 要素と Close Anchor を挿入する。すなわち図4における “[open anchor]” から “[close anchors and inlined document]” への遷移を行う。(2) 非表示になっている Inline Frame の URI 部分 (`src` 属性) を書き替えて、Inline Frame 内に文書ソースを読み込ませる。(3) Inline Frame に読み込んだリンク先文書ソースを `div` 要素の内容にコピーする。以上の処理によって、文書内挿が行われる。

Close Anchor がクリックされたときに呼び出される JScript 関数 `remove_page()` は、ID を設定した `span` 要素について “[close anchors and inlined document]” から “[open anchor]” への遷移を行う。これにより、内挿文書を閉じることができる。

### 3.2 リンクアンカーの自動変換

文書内挿リンクアンカーを実現するには、(1) Web 文書が `inlineLink` のスクリプト (文書内挿のための JavaScript 関数定義) をロードしていることと (2) リンクアンカーが `inlineLink` のスクリプトを呼び出すことの2つが必要となる。

あらかじめコンテンツ制作者がサーバにある文書ソース内のアンカーを書き替えておくことによってこれらの条件を満たすことは可能である。しかし、我々はサーバ内の文書ソースには変更を加えず、同時に文書ソースは可能な限り簡潔な記述にしておくことが保守性および利用者の利便性という点において重要であると考えている。よって、読者がリンクアンカーをクリックしたときに動的に文書ソースを変更する以下の2つの方法を用意した。

```
[original anchor]
<a href="http://www.google.com/">
google </a>
```

```
[open anchor]
<span id="_google">
<a href="javascript:insert_page('_google',
'http://www.google.com/');">
google
</a></span>
```

```
[close anchors and inlined document]
<span id="_google">
<a href="javascript:remove_page('_google',
'http://www.google.com/');">
google! </a>
<div id="_google-div">
<!-- the contents will be replaced as target page source -->
</div>
<a href="javascript:remove_page('_google',
'http://www.google.com/');">
(close) </a>
</span>
```

図4 アンカータグの書換え

Fig.4 Rewriting of an HTML anchor tag

#### 3.2.1 コンテンツ制作者が準備する場合: `inlineLink` のスクリプトによる変換

我々が実装した `inlineLink` のスクリプトには、読み込んだ文書ソースを `div` 要素にコピーする際に、含まれるリンクアンカーを動的に文書内挿リンクアンカー (Open Anchor) に変換する機能を備えている。この方法を利用して読者に文書ソースを変換させる場合には、コンテンツ制作者は以下の作業を行う。(1) 文書内挿機能を付加したい文書のインデックスページへのリンクを含む「メタインデックスページ」を作成する。(2) 作成した「メタインデックスページ」のリンクを文書内挿リンクアンカー (Open Anchor) に書き替える。

読者は `inlineLink` の機能が必要な場合はコンテンツ制作者が用意したメタインデックスページの Open Anchor を選択する。するとリンク先文書ソース内にあるリンクアンカーは `inlineLink` のスクリプトによって自動的に Open Anchor に変換される。よって、以後内挿文書内のアンカーをクリックした場合には再帰的に内挿が行われる。読者は `inlineLink` の機能を必要としない場合には、通常のインデックスページを指定すればよい。

#### 3.2.2 読者が行う場合: サーブレットによる変換

3.2.1 節の方法では、コンテンツ制作者が文書内挿のためのメタインデックスページを用意していない場



図 5 inlineLink Converter: ユーザが入力した URI が指す文書のアンカーを inlineServlet が変換する

Fig. 5 inlineLink Converter: inlineServlet converts anchors of the requested URI page

合には変換できない。我々は、読者が外部のサイトにあるコンテンツに対して inlineLink を適用できるようにするために、任意のサイトにある文書ソースを文書内挿リンクアンカーに書き替える Java サブレット (Servlet) を実装した。このサブレット (inlineServlet) は、要求された URI の文書ソースを取得し、Java の HTMLParser によって解析したのち、アンカーに含まれる URI を inlineServlet への要求の形式に変換した文書内挿リンクアンカーに変換する。要求された文書ソースはすべて inlineServlet を経由する。

inlineServlet はインターネット上に公開されたサーバが利用できるほか、読者の計算機に直接インストールして利用することも可能である。一般的なサブレットコンテナとして実装してあるため、環境を選ばない。

inlineServlet の動作について述べる。読者はサブレットを起動するための文書 (図 5) を表示し、フォームに URI を入力して [convert] ボタンを押すと、URI が指す文書が inlineLink に変換されて表示される。具体的には、各アンカーの URI はサブレットへの要求として変換され、文書ソースは inlineServlet によって取得・変換される。

## 4. 実験

文書内挿による Web 閲覧の効果を測定するため、以下の 3 つの実験を行った。それぞれの実験では、文書の規模 (タスク完了に要求される文書量) が異なる文

書を対象とした。実験では XGA ディスプレイと、スクロールホイール付きのマウスを使用した。

### 4.1 実験 1 (Direc)

中規模の文書例として、Emacs の Directory Editor (Direc) 操作ガイド (全 16 ページ) を準備した。この文書はインデックスページから 15 のページへのリンクが張られている。被験者には Direc の操作に関する問題 15 問を操作ガイドを閲覧しながら解答してもらった。全てのリンクを inlineLink を用いて展開した場合の文書は、標準フォントサイズを使った場合、XGA のディスプレイを使用したときの標準的なウィンドウ高さ (約 630 ピクセル) の約 18 倍になる。

工学系の学部生及び大学院生 15 名からなる被験者を 2 つのグループ (A) (B) に分け、表 1 に示す順序 (1)–(4) にて実験を行った。実験 1 では、被験者は同じ問題と同じ文書を条件を替えて 2 回実験する。被験者は Internet Explorer 5.5 を用い、[normal] では標準機能のみ、[inline] では文書挿入機能 (内挿時の自動位置調整機能なし、ダブルクリックによる文書を閉じる機能なし) を用いて閲覧した。グループ別に実験順を替えることにより一方が有利となる効果は相殺される。全ての問題に解答するのに要したマウスボタンのクリック数、ポインタの移動距離と、作業時間を計測した。

表 1 実験 1 におけるタスクの順序

Table 1 Order of tasks in experiment 1

実験	タスク	グループ (A)	グループ (B)
(1)	問題 1-8 [1st]	normal	inline
(2)	問題 9-15 [1st]	inline	normal
(3)	問題 1-8 [2nd]	inline	normal
(4)	問題 9-15 [2nd]	normal	inline

### 4.2 実験 2 (Glossary)

小規模な文書の例として、HTML で記述されたコンピュータ用語集を準備した。この用語集の本文中に現れる「見出し語」はアンカーによって参照できる。インデックスは利用できないようにし、階層構造を持たない文書として実験を行った。被験者には「ISO と IEC のうち、早く設立された機関を答えなさい。」など、2 つ以上の用語を参照しないと解答できない問題を与えた。被験者には、解答に必要な見出し語へのリンクが全て表示されている状態から閲覧および解答を

GNUjdoc によって配布されている Emacs の操作ガイドを texi2html を用いて texinfo 形式から HTML 形式に変換したのち、Directory Editor の部分を抽出した。  
<http://openlab.ring.gr.jp/gnujdoc/cvsweb/cvsweb.cgi/gnujdoc/emacs-20.6/>

開始してもらった。解答に必要な文書を **inlineLink** を用いて展開した場合の文書は、実験 1 と同じ条件においてウィンドウ高さの約 6 倍以下である。

工学系の学部生及び大学院生 12 名からなる被験者を 6 つのグループに分け、3 つの問題群を実験順を替えて行った。被験者は Internet Explorer 6.0 を用い、[normal] では標準機能のみ、[inline] では文書挿入機能（内挿時の自動位置調整機能なし）、[inline(adjust)] では文書挿入機能（内挿時の自動位置調整機能あり）を用いて閲覧してもらった。[inline] および [inline(adjust)] においては、ダブルクリックによる文書を閉じる機能を有効にした状態で閲覧してもらった。全ての問題に解答するのに要したマウスボタンのクリック数、ポインタの移動距離、ドラッグ距離（ボタンを押した状態でのポインタ移動距離）、ホイール回転量、作業時間を計測した。

#### 4.3 実験 3 (TBL)

比較的大規模な文書の例として、Tim Berners-Lee による Web ガイドライン “Style Guide for online hypertext” の日本語翻訳版 を用いた。この文書は 28 ページから構成されており、インデックスから辿れる 23 のページのほか、文書中の記述から辿ることのできる補助的なリンクを備えている。全てのリンクを **inlineLink** を用いて展開した場合の文書は、実験 1 と同じ条件においてウィンドウ高さの約 50 倍になる。

実験 2 と同じく 12 名の被験者を 6 つのグループに分け、3 つの問題群について [normal]、[inline]、[inline(adjust)] を順序を替えて閲覧しながら解答してもらった。問題 1、2 は決められた文章を文書内から見つけるタスクを、問題 3 は「タイトルの長さは何文字以内にするのが望ましいか」というように、文書の内容を理解して答える種類のタスクを与えた。実験 2 と同じく [inline] および [inline(adjust)] においては、ダブルクリックによる文書を閉じる機能を有効にした状態で閲覧してもらった。全ての問題に解答するのに要したマウスボタンのクリック数、ポインタの移動距離、ドラッグ距離、ホイール回転量、作業時間を計測した。

#### 4.4 結果と考察

まず、観察によって得られた被験者の行動について述べる。実験において、文書内挿を用いた場合のウィンドウ高さに対する文書の長さは、個人差はあるが最大で [Dired] では 5~10 倍、[Glossary] では 2~3 倍、[TBL] においては 10~20 倍程度の長さが観測された。[normal] の場合には、被験者にはマウスを用いて行

うブラウザの機能（新しいウィンドウを開く操作とウィンドウ調整、アンカーのドラッグ&ドロップ）を自由に利用してもらった。その結果 [Glossary] において、新しいウィンドウを開く操作を行う被験者が半数程度見られた。新しいウィンドウを 1 つだけ開き、アンカーのドラッグ&ドロップによって閲覧した被験者は順調にタスクを完了した。しかし、ウィンドウを 2 つ以上開いてしまった被験者は、ウィンドウの調整に手間取ったり、多数の類似したウィンドウを扱っているうちに元のウィンドウを見失ってしまうといった現象が見られた。

次に、実験によって得られたデータについて議論する。図 6、図 7、図 8 に、各実験におけるマウスクリック数、移動距離、作業時間の平均および標準誤差を示す。表 2 に、片側有意水準 5% にて  $t$  検定を行った結果 ( $t$ ) を示す。括弧のない数字は平均値に有意な差があると判定された項目である。+ は「比較した手法」の左側の手法に比べて右側の手法を用いた場合減少することを表し、- は増加することを表している。

[Dired] と [Glossary] については、**inlineLink** ([inline] もしくは [inline(adjust)]) を用いることによってマウスクリック数と移動量は有意に減少した。[TBL] については、[inline] においてマウス移動量のみが有意に減少している。[TBL] においてクリック数が減少していない理由として、ダブルクリックによって文書を閉じる操作を、被験者が頻繁に利用したことが挙げられる。特に、[inline(adjust)] の場合、内挿時の自動調整によって Open Anchor とマウスポインタの位置がずれてしまうため、特にダブルクリックによる操作が頻繁に観測された。[Dired] については、ダブルクリックによって文書を閉じる機能は無効であったので、被験者は文書を閉じる際には、Open Anchor または Close Anchor までマウスを移動していた。また、スクロール操作にホイールを使用せず、スクロールバーを直接操作する被験者も確認された。そのため、[Dired] のマウス移動量は [Glossary] と比べるとそれほど減少していない。[Glossary] の場合、クリック数、マウス移動量共に減少した。この原因として、[Glossary] は [Dired] や [TBL] に比べて内挿される文書が短いため「内挿した文書を閉じる」という操作がそれほど観測されなかったことが挙げられる。

作業時間の平均値は、**inlineLink** を用いると減少する傾向にあるが、有意差は観測できなかった。この理由として、作業時間には操作時間のほかに、被験者が迷ったり、文書を読解するというタスク遂行のために必要な時間が多く含まれるため、純粋な操作時間の有

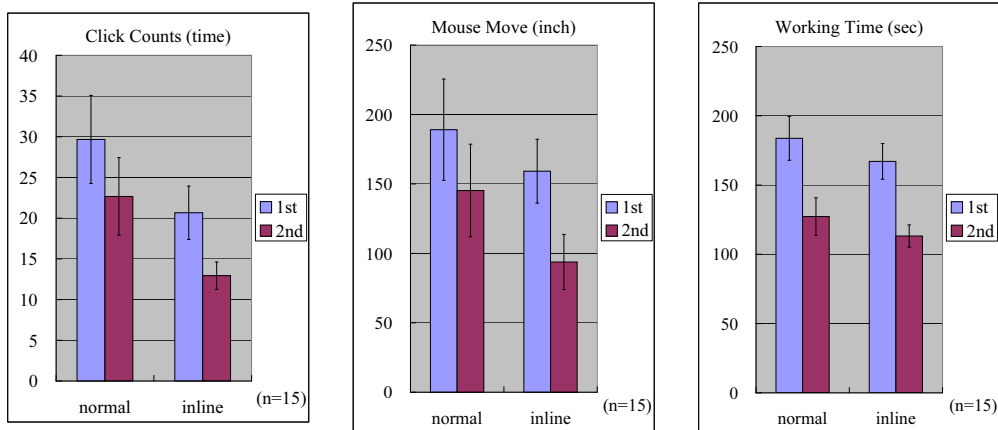


図 6 実験 1 (Direc) の結果

Fig. 6 Result of experiment 1 (Direc)

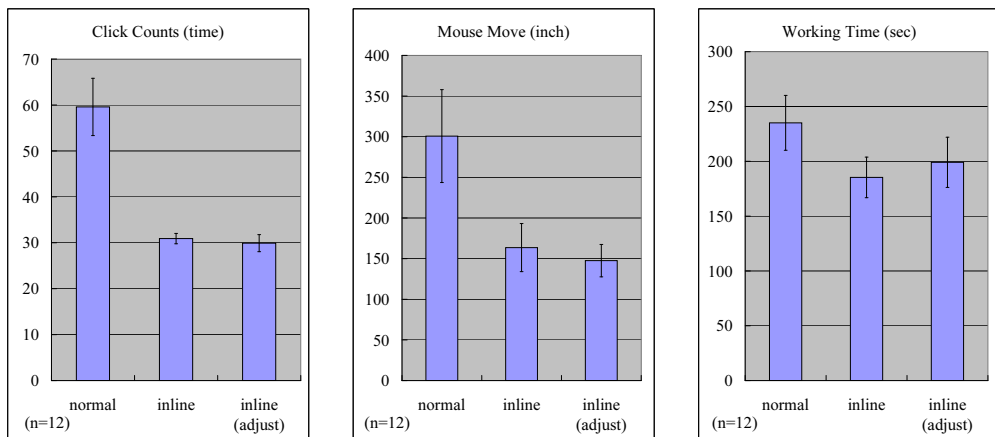


図 7 実験 2 (Glossary) の結果

Fig. 7 Result of experiment 2 (Glossary)

表 2 t 検定結果

Table 2 Result of t-test

実験	比較した手法	mouse click	mouse move	time	wheel up	wheel down	mouse drag	片側 5% 境界値
Direc	normal-inline	+2.78	+2.24	(+1.04)				1.70
Glossary	normal-inline	+4.52	+3.47	(+1.65)	-2.40	-4.37	+3.05	1.80
Glossary	normal-adjust	+4.62	+3.31	(+1.15)	(-1.60)	(-1.72)	+2.96	1.80
Glossary	inline-adjust	(+0.32)	(+0.71)	(-0.46)	(+0.53)	+2.12	(-0.89)	1.80
TBL	normal-inline	(+0.93)	+2.12	(+1.12)	(-0.36)	(+0.62)	(+1.74)	1.80
TBL	normal-adjust	(-0.26)	(+1.26)	(+0.04)	-2.52	(+0.30)	+2.38	1.80
TBL	inline-adjust	(-1.38)	(-1.66)	(-1.41)	-2.06	(-0.45)	(+1.30)	1.80

意差が現れにくいことが考えられる。

[Glossary] と [TBL] について計測したホイール回転量と、ドラッグ距離について考察する。ホイール回転量の計測によって、[Glossary] においては [inline(adjust)] の自動調整機能がホイールを用いた下方方向のスクロール量を軽減していることが分かった。しかし、一般に inlineLink を用いて文書内挿を行う場合はホイール回

転量は増加する傾向にある。この傾向が [Glossary] で特に顕著に現れたのは、[Glossary] に用いた文書が短く、[normal] においてスクロールが行われなかったためである。[TBL] のような長い文書の場合は、[normal] においてもスクロールが必要となり、ホイールが利用される。[TBL] の場合、[inline] におけるホイール回転量の増加は [normal] と比べて有意差が現れない程度に



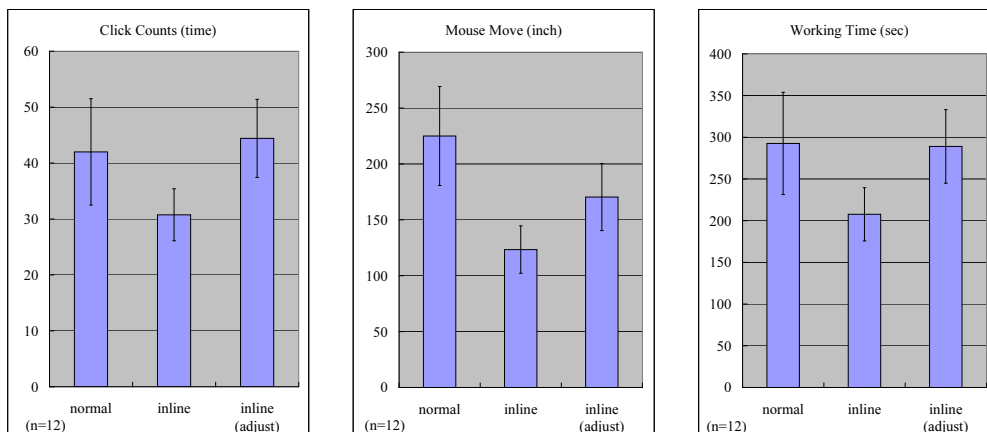


図 8 実験 3 (TBL) の結果  
Fig. 8 Result of experiment 3 (TBL)

抑えられている。ただし、自動調整機能を用いた場合には上方向のホイール回転量が増加している。この原因は、実験に用いた自動調整機能が不十分であった点にあると考えられる。実験に用いた自動調整機能は、内挿時のみ下方向へのスクロールを行うもので、文書を閉じる際に上方向へのスクロールは行わない。そのため、被験者は 1 つの文書を開閉した後に元の位置に戻す際、自動調整機能がスクロールした分だけ上方向スクロールを行う必要があった。

実験の結果、内挿される文書の長さがウィンドウの高さに比べて短い場合には文書内挿機能は有効に機能し、マウスのクリック数と移動量を軽減する効果があると言える。また、内挿時の自動調整機能がホイールによる下方向のスクロール量を軽減する効果が確認できた。

実験後、被験者に行ったアンケートには、「文書を閉じた時の自動位置調整機能が必要」というコメントがある一方で、「自動位置調整機能は分かりにくいので不要」というコメントも若干見られた。また、ダブルクリックによって文書を閉じる機能については、必要であると考えの人が多かったが、「アンカー以外の場所で閉じる機能は必要だがダブルクリックに割り当てるのは最善かどうか分からない」という意見もあった。inlineLink の有効性に関しては、「内挿される文書が短い場合には読みやすい」「必要な場面において使えるようになっていけば便利ではないか」というコメントが多かった。

## 5. 議 論

### 5.1 読者の利点

文書内挿リンクを Web 文書閲覧に適用する利点と

して、読者にリンク元文書を常に意識させることにより知らないうちに本来の作業と関係ない文書を閲覧してしまうことを防止する効果もある。また、内挿した文書を含めて 1 つの文書として扱えるため、キーワードによる検索を一括して行うことができる。さらに、内挿した文書を印刷することにより、画面に表示されている状態（複数のリンク先が内挿されている状態）を紙に出力することが可能である。

Tauscher らによる Web 読者の行動調査<sup>8)</sup>では、Web ブラウザを用いたナビゲーション行為のうち約 40%が「アンカーをクリックしてリンクをたどる行為」であり、約 30%が「“Back” ボタンによる再参照」であると述べている。実際の閲覧においては、これらの行為を交互に行う場面は少なくない。この 2 つの行為を繰り返し行う場面において、inlineLink はマウスポインタを移動することなく実行できる点で効果的であるといえる。

文書内挿を繰り返した場合、一般に文書は長大になるため、一覧性は低下し、移動時のスクロール時の負荷は増加する。しかし、「文章を読解する状況」においては、文書間の関連やつながりを自然に表現する文書内挿の特長は文書が長大になった場合にも失われない。長大な文書のスクロールは、読者に負担を強いる操作であるが、通常の記事閲覧時に行われている操作であり、近年普及しているホイール付きマウスを用いることによりポインタや視点を移動する必要がないため文章を読む作業への影響は軽減される。また、2.5.2 節で述べた「注目点に応じた位置合わせ」や「インデックスによる移動」といった機構を用いることによりさらに負荷は軽減できると考えられる。

## 5.2 文書作成者の利点

**inlineLink** は読者だけでなく、文書作成者にも利点がある。2.5.2 節で述べたように、**inlineLink** は文書間の移動を支援する。文書間の順序を示すリンクのインデックスが存在する場合、**inlineLink** の機能によって、ページ間移動を行うことができる。そのため、文書作成者は [Previous][Next][Top] 等の文書間移動専用のリンクアンカーをあえて追加する必要はない。これらを追加しない場合には、文書構造を変更する作業においても修正作業が軽減される。

また、3.2.1 節で述べた **inlineLink** のスクリプトによるリンクアンカーの変換機能を用いることによって、文書制作者は内挿機能を付加したい文書のインデックスページへのアンカー (Open Anchor) を 1 つ準備するだけで済む。各文書のリンクアンカーを修正する必要がないため、元の Web 文書を簡潔な状態に保つことができる。そのため、展開機能の有無を読者が選択しやすい。このことは、Microsoft の Web サイト等にて提供されている展開機能に比べると、より自由度の高い状態で実現していると言える。

なお、内挿文書の範囲を明示するための視覚効果は、CSS (Cascading Style Sheet) によって簡単に変更できる。スタイルシートを数行変更するだけで枠で囲うスタイル (図 1) とインデントスタイル (図 2) とを切り換えたり、色を変更することができる。また、**inlineLink** では閲覧時に用意したスタイルシートを切り換える機能を備えているため、スタイルや文字フォントを読者の好みによって変化させることができる。

## 6. 関連研究

Web 文書閲覧の際に発生するナビゲーションに関する負荷を軽減する研究は、大きく 2 つに分類できる。1 つはリンク先文書に関する情報を少ない操作量で参照することにより読者がそのリンクをたどるべきか否かを判断する材料を事前に提供する研究である。もう 1 つは Web 文書を閲覧する際の操作を簡略化する機構に関する研究である。

### 6.1 リンク先情報の提示

Visual Link Preview<sup>9)</sup> は、リンク先文書の情報を先読みし、縮小画像 (サムネイル) を生成して表示する。HyperScout Linktool<sup>10)</sup> は、リンク先文書の情報 (タイトル、著者、言語、前回の訪問日時、サーバの反応時間等) をテキストとアイコンによる表現を用いてユーザに提示する。どちらもアンカーにマウスポインタを合わせたときにポップアップウィンドウを用いて元のアンカーの近くに表示するため、読者の視点移動を減

らすよう考慮されている。これらの情報は読者がリンクをたどる際の判断材料としては有効であるが、リンク先文書とリンク元文書を同時に表示することはできない。**inlineLink** では、内容に連続性のある複数の文書を同時に可読な状態で表示することができる。

Fluid Link<sup>11),12)</sup> では、注釈などの付加情報 (gloss) を、元の文章のアンカーの近くに配置するためのアニメーションや半透明表示などを用いた高度な表現方法を提案している。Fluid Link は専用のドキュメントブラウザで実現されているため、**inlineLink** よりも表現能力は高い。**inlineLink** は可搬性を重視し、Fluid Link に似た文章閲覧インタフェースを従来の Web ブラウザ上で実現した。

### 6.2 閲覧動作の簡略化

Elastic Windows<sup>13),14)</sup> は、指定した複数のアンカーを一度に開いて閲覧することができる Web ブラウザである。リンク先文書はそれぞれリンク元文書の右側に配置され、読者は文書構造を把握しながら閲覧することができる。また、複数ウィンドウの開閉や移動操作を同時に実行できるため、階層を持つ文書の一貫性と操作性の向上に貢献している。ただし、アンカーとリンク先文書との位置の対応はタイトルバーの文字列によって照合するか、読者が記憶していなければならない。文書内挿を用いた場合には、アンカー位置の直後にリンク先文書が配置されるので、読者が位置を記憶する必要はない。

Zero-Click<sup>15)</sup> は **inlineLink** と同様に閲覧操作の軽減を目的としたシステムである。読者がアンカーにポインタを合わせるによりリンク先の文書がポップアップウィンドウに表示される。Visual Link Preview におけるサムネイルに比べてリンク先文書の情報を多く表示できる反面、元の文書を覆ってしまうウィンドウの面積が広くなるという問題がある。

SmallBrowse<sup>16)</sup> は、Web の閲覧履歴を用いて読者が次にたどるリンクを推測する PBE (Programming by Example) システムである。PDA などの比較的小さな画面において、スクロールしながらリンクを探す手間を抑えることを目的としている。SmallBrowse では、提示したアンカーリンクに関する情報を TipHelp というポップアップウィンドウで表示する。閲覧動作における負荷を軽減するという点では **inlineLink** と共通している。SmallBrowse は定期的にページを訪問し閲覧する際に効果を発揮する。**inlineLink** は構造を持つ文書群に対して読者が表示する情報の粒度を調節できる。

## 7. ま と め

Web 閲覧において文書内挿リンクを利用するためのシステム **inlineLink** について述べた。 **inlineLink** では、ナビゲーション機能を強化することにより文書内挿における移動の負荷を軽減した。 **inlineLink** の実装にあたり、可搬性を重視し一般に普及している Web ブラウザ上で動作するように設計した。 **inlineLink** と通常の Web ブラウザにおける閲覧比較実験を行い、Web 閲覧における文書内挿リンクがマウスのクリック数および移動量を抑えることを確認した。

**inlineLink** および関連スクリプトは以下の URI から利用できる。

<http://www.iplab.is.tsukuba.ac.jp/~miuramo/inlinelink/>

## 8. 謝 辞

査読者の方には、本研究の有効性、適用可能な文書に関して有益なコメントをいただきました。ここに感謝の意を表します。

## 参 考 文 献

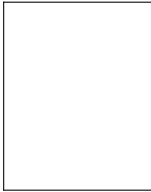
- 1) Motoki Miura, Buntarou Shizuki, and Jiro Tanaka. inlineLink: Inline Expansion Link Methods in Hypertext Browsing. In *Proceedings of 2nd International Conference on Internet Computing (IC'2001)*, Vol. II, pp. 653–659, June 2001.
- 2) Peter J. Brown. Turning Ideas into Products: The Guide System. In *Hypertext'87 Proceedings*, pp. 33–40, November 1987.
- 3) Jakob Nielsen. *Multimedia and Hypertext: The Internet and Beyond*, chapter 9, pp. 247–278. AP Professional, 1995.
- 4) Jakob Nielsen and Uffe Lyngbæk. Two field studies of hypermedia usability. In Ray McAleese and Catherine Green, editors, *Hypertext: state of the art*, chapter 7, pp. 64–72. Intellect Ltd., 1990.
- 5) Benjamin B. Bederson, Jon Meyer, and Lance Good. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. In *Proceedings of the 13th annual ACM Symposium on User Interface Software and Technology (UIST2000)*, pp. 171–180, 2000.
- 6) Document Object Model (DOM) Level 2 Specification. (Web Page), May 2000. <http://www.w3.org/TR/DOM-Level-2/cover.html>.
- 7) World Wide Web Consortium (W3C). HTML 4.01 Specification. (Web Page), December 1999. <http://www.w3.org/TR/html401/>.
- 8) Linda Tauscher and Saul Greenberg. How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems. *International Journal of Human-Computer Studies*, Vol. 47, No. 1, pp. 97–138, 1997.
- 9) Theodorich Kopetzky and Max Mühlhäuser. Visual Preview for Link Traversal on the WWW. In *Proceedings of the 8th International World Wide Web Conference (WWW8) / Computer Networks 31 (11-16)*, pp. 1525–1532, 1999.
- 10) Harald W. R. Weinreich and Winfried Lamersdorf. Concepts for Improved Visualization of Web Link Attributes. In *Proceedings of the 9th International World Wide Web Conference (WWW9) / Computer Networks 33 (1-6)*, pp. 403–416, 2000.
- 11) Polle T. Zellweger, Bay-Wei Chang, and Jock Mackinlay. Fluid Links for Informed and Incremental Link Transitions. In *Proceedings of HyperText'98*, pp. 50–57, 1998.
- 12) Polle T. Zellweger, Susan Harkness Regli, Jock D. Mackinlay, and Bay-Wei Chang. The Impact of Fluid Documents on Reading and Browsing: An Observational Study. In *Proceedings of the CHI 2000 conference on Human factors in computing systems (CHI'00)*, pp. 249–256, April 2000.
- 13) Eser Kandogan and Ben Shneiderman. Elastic Windows: A Hierarchical Multi-Window World-Wide Web Browser. In *Proceedings of the 10th annual ACM Symposium on User Interface Software and Technology (UIST'97)*, pp. 169–177, October 1997.
- 14) Eser Kandogan and Ben Shneiderman. Elastic Windows: Evaluation of Multi-Window Operations. In *conference proceedings on Human factors in computing systems (CHI'97)*, pp. 250–257, March 1997.
- 15) 南野朋之, 斎藤豪, 奥村学. Web ブラウジング支援システム Zero-Click. *インタラクティブシステムとソフトウェア IX*, pp. 131–136. 近代科学社, December 2001.
- 16) Atsushi Sugiura. Web Browsing by Example. In Henry Lieberman, editor, *Your Wish is My Command – Programming by Example –*, chapter 4, pp. 61–85. Morgan Kaufmann Publishers, 2001.

(平成 2002 年 4 月 15 日受付)


(平成 2002 年 10 月 7 日採録)

**三浦 元喜 (正会員)**

1974年生。1997年筑波大学第三学群情報学類卒。2001年筑波大学博士課程工学研究科修了。博士(工学)。現在、筑波大学電子・情報工学系助手。筑波大学先端学際領域研究(TARA)センター勤務。Webインタフェース、視覚化、グループウェアに興味を持つ。情報処理学会、日本ソフトウェア科学会、ACM各会員。

**志築文太郎 (正会員)**

1971年生。1994年東工大理学部情報科学科卒。2000年同大学大学院情報理工学研究科数理・計算科学専攻博士課程単位取得退学。博士(理学)。現在、筑波大学電子・情報工学系助手。ヒューマンインタフェースに関する研究に興味を持つ。情報処理学会、日本ソフトウェア科学会、ACM、IEEE Computer Society、電子情報通信学会、ヒューマンインタフェース学会各会員。

**田中 二郎 (正会員)**

1951年生。1975年東京大学理学部卒。1977年同大学院理学系研究科修士課程修了。1984年米国ユタ大学計算機科学科博士課程修了、Ph.D. in Computer Science。1993年より筑波大学に勤務。現在、電子・情報工学系教授。2001年より筑波大学先端学際領域研究センター(TARAセンター)実世界指向インタラクション研究プロジェクト研究代表者。プログラミング言語やヒューマンインタフェースに興味を持つ。ACM、IEEE Computer Society、日本ソフトウェア科学会、電子情報通信学会、人工知能学会、ヒューマンインタフェース学会各会員。