

ハイパーテキストにおける文書挿入型 リンク機能の提案

An Inline Expansion Method for Hypertext Link

三浦 元喜[†]
Motoki MIURA

田中 二郎^{††}
Jiro TANAKA

[†]筑波大学 博士課程 工学研究科

Doctoral Program in Engineering, University of Tsukuba

^{††}筑波大学 電子情報工学系

Institute of Information Sciences and Electronics, University of Tsukuba

概要

従来の Web ブラウザを用いてリンクをたどった場合、画面に表示されている元の文書が消されてしまいユーザが読んでいる文脈(コンテキスト)が失われる。我々は、元のアンカー位置にリンク先の情報を埋め込む「文書挿入型リンク機能」を提案する。アンカーを含む元の文書を残しつつリンク先の文書を表示するので、視点の移動を抑えることができる。また、ウィンドウ操作をしなくて済むため、ユーザは閲覧作業に集中しやすい。

1 はじめに

Web ページなどのハイパーテキストを閲覧する行為において、ハイパーリンクをたどる機能は頻繁に利用される。多くの Web ブラウザでは、関連する URI (Uniform Resource Identifier) が定義された文字列や画像 (以下「アンカー」) をマウスでクリックすることでその URI が特定する Web ページを取得して閲覧できる。既存のブラウザは、たどったリンク先の文書を表示するため、元の文書を消してしまう。そのため、ユーザはリンク元の文書とリンク先の文書を同時に閲覧する場合に、特殊な操作を行う必要があった。

そこで我々は、ハイパーリンクをたどるときの新しい表示方法として、埋め込み型リンク手法を提案する。本手法を用いると、ユーザはリンク元の文書とリンク先の文書を同時に閲覧できる。

2 既存のブラウザにおけるリンクをたど

る操作とその問題点

Web ブラウザにおいてリンクをたどる操作には、以下の3つがある。それぞれの操作を説明し、その問題点を指摘する。

操作1: アンカーをクリックする

アンカー上でマウスボタンをクリックする方法。元の文書が表示されていたウィンドウに、リンク先の新しい文書が表示される。このとき、元の文書が消えてしまうので、ユーザが今まで読んでいた文脈(コンテキスト)が失われてしまうという問題がある。

操作2: “新しいウィンドウで開く” メニューを選択する

アンカー上でメニューを開き “Open Link in New Window” を選択する方法。新しく開いたウィンドウにリンク先の文書が表示される。リンク元の文書は保存されるが、ウィンドウの操作が必要になることが多い。例えば、新しく開いたウィンドウが元の

文書のウィンドウと重なってしまった場合、移動やリサイズをしなければ元の文章が参照できない。このような操作は文書を読む作業を中断させるため、効率が悪い。

操作3：アンカーをドラッグ&ドロップする
複数のウィンドウが表示されている状態で、アンカーをドラッグし、別のウィンドウにドロップする方法。この操作により、ユーザがリンク先の文書を表示させるウィンドウを直接指定できる。しかし、どのリンクをどのウィンドウに表示したかという情報は画面に残らないため、ユーザが記憶しておく必要がある。

3 提案手法 “inlineLink”

既存のブラウザにおいて発生する上記の問題を解決するために、我々はリンクされたハイパーテキストを参照するための新しい手法 “inlineLink” を提案する。inlineLink とは、従来個別にウィンドウを割り当てて表示していたリンク先文書を、リンク元文書の内部に埋め込んで表示する手法である。図1 に、inlineLink の概念図を示す。リンク先の文書は、その文書へのリンクを持つアンカーの近くに配置する。元の文書においてアンカーより後にあった文書は、内挿されたリンク先の文書の後に配置する。ユーザは、必要に応じてそのリンク先文書を閲覧し、不要になった時点でその文書を取り除くことができる。inlineLink では、展開した文書中のリンクをさらに展開することで、多段階の入れ子にすることも可能である。

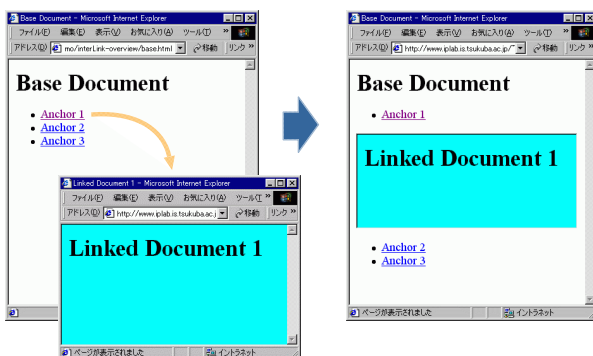


図1 従来の手法(左)と、inlineLink 手法(右)

inlineLink の利点を述べる「操作1」でリンクをたどった場合では、元の文書に戻るために “Back”

ボタンを用いる必要があったが、inlineLink では、挿入された文書の下に元の文書が続いているので、下方向にスクロールするだけで読む作業を継続できる。スクロール操作は、通常の文書の閲覧時に行われている操作であり、文章を読む作業への影響は少ない。また「操作2」で発生するウィンドウの移動・リサイズ操作は必要ない。さらに「操作3」ではユーザがアンカーとウィンドウの位置関係を覚えておく必要があったが、inlineLink ではアンカーの近くに表示するのでアンカーとウィンドウの対応付けが容易である。また「操作2,3」では、別ウィンドウに表示された文書を閲覧するためにユーザの視点はウィンドウ間を移動するが、inlineLink ではリンク先の文書をアンカーの近くに表示するので、文書を閲覧するのに必要な視点移動距離は少なくなる。特にこれらの文書間を交互に参照する場面で有効である。

4 実現

Webブラウザ上でのinlineLink機能の実現について述べる。

4.1 方針

我々は、Webブラウザ上でinlineLinkを実現するために、Dynamic HTML [1] による動的なページ変更機能を用いる。Dynamic HTMLを用いた理由として、一般に広く利用されているWebブラウザ上で動作することが挙げられる。inlineLinkの機能をブラウザのリンク機能を拡張して実装する場合に比べ、ユーザの環境やブラウザを制限しないという点で優れている。

Dynamic HTMLでHTMLオブジェクトをコントロールするスクリプト言語としては、VBScriptやJavaScriptがある。我々は、プラットフォームに依存しないという点でJavaScriptを採用した。

4.2 リンク先文書の挿入方法

Dynamic HTMLを用いてinlineLinkを実現する方法として以下の2つが考えられる。

[Type-1] HTMLで記述されたソースを挿入する方法

リンク先のHTML文書のソースを、リンク位置に埋め込む方法である(図2)。例えば、<body> タグ

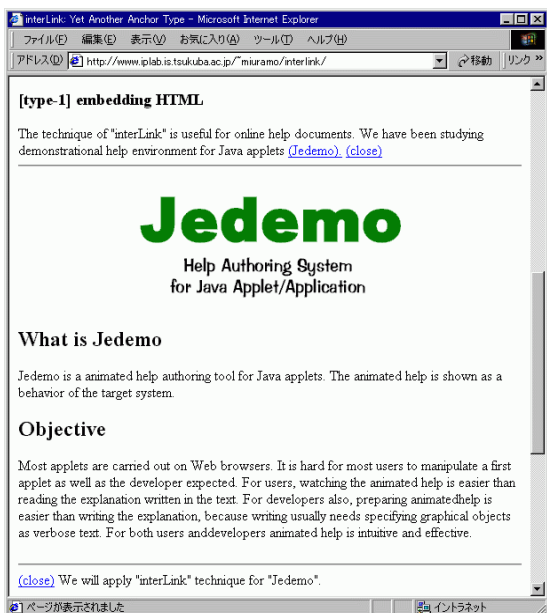


図2 [Type-1] HTML で記述されたソースを挿入

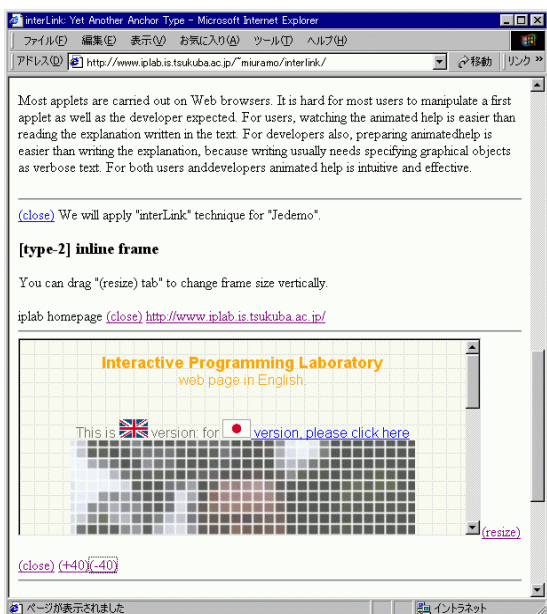


図3 [Type-2] Inline Frame を用いて挿入

で囲まれた部分 (body 要素の内容) を切り出して、リンク位置の直後に挿入する。[Type-1] では、挿入される文書が長い場合にリンク位置より後にあった元の文書が画面から失われる可能性がある。また、言語や文字コードが混在する、背景色が保存されない* などにより、うまく表示されない恐れがある。

[Type-2] Inline Frame を用いる方法
リンク先の HTML 文書を、リンク位置に挿入し

* CSS (Cascading Style Sheet) などのスタイルシートを用いることで、背景色を変更できる。

た Inline Frame の中に表示する方法である (図3)。Inline Frame は、HTML4.0 Transitional DTD [2] で iframe 要素として定義されているもので、文書中にオブジェクトとして挿入できるフレームである。Inline Frame のサイズは自由に設定できるため、挿入される文書が長い場合でも一部分だけを表示可能である。さらに、背景色を保存するので [Type-1] に比べると本来の画面に近い。

4.3 HTML の書き替えによる実現

Dynamic HTML を用いた inlineLink 機能の実現法について述べる。HTML 文書に含まれるアンカータグ (a 要素) の後に、インライン要素である span 要素を挿入する (図4)。この span 要素は、内容を書き替えるときに特定するため id 属性を設定しておく。また、リンク先文書を展開するときをクリックするアンカーを設定しておく。このアンカー上でクリックすると、指定された id を持つ span 要素の内容自身を書き替える JavaScript の関数が行われる。insert_page 関数は、第1引数で指定された id を持つオブジェクト (レイヤ) に、第2引数で与えられた URI のページを表示するように我々が定義した関数である。この関数が実行されると、span 要素の内容は図5のように書き替えられ、リンク先の文書が span 要素の位置に挿入されて表示される。ちなみに、図5では、iframe 要素を挿入する例を示しているが、同様の手法で表や画像などの HTML オブジェクトを挿入できる。

```

    本来のA要素
    <a href="http://www.google.com">google</a>
    <span id="_google"> 挿入されたSPAN要素
    <a href="javascript:insert_page('_google',
    'http://www.google.com/')">(open)</a>
    </span>
  
```

図4 埋め込んでおく span 要素の例

```

    本来のA要素
    <a href="http://www.google.com">google</a>
    <span id="_google"> 挿入されたSPAN要素
    <a href="javascript:remove_page('_google');">
    (close)</a>
    <hr>
    <iframe src="http://www.google.com/"
    width=90% height=300>
    http://www.google.com </iframe> IFRAME要素
    <hr>
    <a href="javascript:remove_page('_google');">
    (close)</a>
    </span>
  
```

図5 展開された後の span 要素の例

4.4 フィルタプログラム

我々は、HTML 文書中にある a 要素の後に、上で示したような span 要素を埋め込むフィルタプログラムを Perl で実装した。HTML ファイルを入力すると、inlineLink 機能を追加した HTML ファイルを出力する。

4.5 フィルタプログラムの使用方法

Web ページ製作者が使用する場合とユーザが使用する場合がある。Web ページ製作者は、このフィルタプログラムを使って inlineLink 機能をあらかじめ追加しておくことができる。

一方、ユーザは Web ブラウザを使うときに、inlineLink 対応ページに変換するプロキシサーバを通して通常の HTML 文書を書き替えることができる。

4.6 ユーザが可能な操作

通常のリンクの直後に、“(open)” というアンカーを表示する。これをクリックすることで、文書が書き替えられてリンク先の文書が表示される。

リンク先の文書が表示された後では、“(close)” というアンカーをクリックすることでリンク先の文書を閉じることができる。[Type-2] においては、“(resize)” をドラッグしたり、“(+40)” や “(-40)” をクリックすることで Inline Frame のサイズを調整できる。内挿された文書の URI はフレームの上部に表示しており、通常のリンクと同様に機能する。ちなみに、“(resize)” などの文字列の代わりに、画像を用いることも可能である。

5 関連研究

リンク先の情報をユーザに示唆する目的で、アンカーにマウスポインタを合わせたときにサムネイル画像 (Visual Preview) [3] や、gloss と呼ばれる付加情報 [4] を提示する研究がある。どちらも情報を元のアンカーの近くに表示してユーザの視点の移動量を減らすよう考慮している。これらの研究の目的は、ユーザが多数のリンクの中から重要なリンクを選択するのに有益な情報を提示することにある。我々は、リンク先の文書をすべて可読な状態で提示

することに主眼を置いている。

リンク先の文書を可読な状態で表示する研究として、複数のウィンドウを適切に配置する Elastic Windows [5] がある。元のページを表示したままリンク先ページを開くという点では、「操作 2」の拡張といえる。ウィンドウ操作を簡略化できるので、階層をもつページの一覧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、ユーザが記憶していなければならない。我々の手法を用いた場合では、アンカー位置の直後にリンク先文書が配置されるので、ユーザが位置を記憶する必要はない。

6 まとめ

リンク先の文書を元の文書に埋め込む手法 inlineLink を提案した。本手法を用いると、従来の Web ブラウザでリンクをたどる操作をした場合の問題を解決できる。一般的なブラウザで動作させるため、Dynamic HTML の技術を用いた実現法を示した。

ナビゲーション機能を充実させることと、本手法の有効性を検証することが今後の課題である。

参考文献

- [1] Microsoft Corporation. DHTML および DHTML オブジェクト モデルについて - Microsoft Office 2000/Visual Basic プログラマーズガイド - <http://www.microsoft.com/JAPAN/developer/library/odeopg/deovrworkingwithdhtmlhtmlobjectmodel.htm>.
- [2] World Wide Web Consortium (W3C). HTML 4.01 Specification. <http://www.w3.org/TR/REC-html40/>.
- [3] Theodorich Kopetzky and Max Mühlhäuser. Visual Preview for Link Traversal on the WWW. In *Proceedings of the WWW8*, 1999. <http://www8.org/w8-papers/4b-links/visual/visual.html>.
- [4] Polle T. Zellweger, Bay-Wei Chang, and Jock Mackinlay. Fluid Links for Informed and Incremental Link Transitions. In *Proceedings of HyperText'98*, pages 50-57, 1998.
- [5] Eser Kandogan and Ben Shneiderman. Elastic Windows: A Hierarchical Multi-Window World-Wide Web Browser. In *Proceedings of UIST'97*, pages 169-177, 1997.