

平成17年度

筑波大学第三学群情報学類

卒業研究論文

題目 Web ページの概観と更新差分を用いた
ブラウジング支援

主専攻 情報科学主専攻

著者 若松亮太

指導教員 田中二郎 志築文太郎 三末和男 高橋伸

要 旨

Web 上では，情報の追加・削除などの更新が頻繁に行われている．また，ユーザが閲覧する Web ページの 81% は，そのユーザによって過去に閲覧されていることが分かっている．したがって，再訪問時に更新が行われている状況が頻繁に見られる．このことから，更新情報を Web ページに併せて把握しやすく表示することで，再訪問時にユーザの情報収集支援を与える．

本研究では，Web ページへの再訪問時の情報収集支援を目的として，ユーザが訪問している Web ページ内の情報を蓄積し，再訪問時に更新差分を提示するシステムを提案する．さらに，更新差分を Web ページの概観と併せてユーザに把握しやすく提示するインタフェースを提案する．提案したインタフェースを実現するシステム StackOverview について詳細を述べるとともに，その活用例を述べる．

目次

第 1 章 序論	1
第 2 章 Web ブラウジングと再訪問の現状	3
2.1 再訪問	3
2.2 再訪問時の情報収集	3
2.2.1 情報の分類	3
追加情報	4
注目情報	4
削除情報	4
2.2.2 再訪問時の問題	4
更新による Web ページの変化	4
情報の位置認識とアクセス	5
第 3 章 Web ページの概観と更新差分	6
3.1 概観提示	6
3.2 情報の提示	6
3.2.1 追加情報 / 注目情報	6
3.2.2 削除情報	8
削除情報 / リンクの展開	8
第 4 章 StackOverview の実装	10
4.1 システムの概要	10
4.2 概観提示	11
4.2.1 縮小画像	11
4.2.2 スクロール機能	11
4.3 情報の蓄積	12
4.3.1 Web ページの自動収集	12
4.3.2 注目情報のスクラップ	12
4.4 情報の抽出	13
4.4.1 追加情報	13
4.4.2 注目情報 / 削除情報	14
4.5 情報の提示	14

4.5.1	注目情報	14
4.5.2	追加情報	15
4.5.3	削除情報	15
4.5.4	inlineLink によるリンクの展開	16
第 5 章	システムの活用例	17
5.1	ウェブログの記事	17
5.2	ニュースサイトの記事	18
第 6 章	関連研究	19
第 7 章	結論	20
	謝辞	21

図目次

3.1	Web ページの概観	7
3.2	追加情報 / 注目情報の提示	7
3.3	削除情報の提示	8
3.4	リンク / 削除情報の展開	9
4.1	Web ページの概観	11
4.2	コンテキストメニューからのスクラップ	13
4.3	注目情報の提示	14
4.4	追加情報の提示	15
4.5	削除情報の提示	16
4.6	inlineLink によるリンクの展開	16
5.1	従来のブログ再訪問	17
5.2	StackOverview によるブログ再訪問	17
5.3	従来のニュースサイト再訪問	18
5.4	StackOverview によるニュースサイト再訪問	18

表 目 次

4.1	一般的なスクロールバーの機能	12
4.2	StackOverview のスクロール機能	12
4.3	強調される情報と色	15

第1章 序論

World Wide Web（以下 Web）上では、情報の追加・削除などの更新が頻繁に行われている。また、Andy Cockburn らの調査 [1] によると、Web ブラウジングにおいてユーザが閲覧する Web ページの 81% は、そのユーザによって過去に閲覧されている。すなわち、Web ブラウジングの大部分は**再訪問**であり、再訪問時に更新が行われているという状況が頻繁に見られる。ユーザがその Web ページの過去の情報を持っているときに必要とする情報は、過去に重要であると認識した情報はまだ存在するか、どのような情報が追加されたか、それらの情報はどこにあるかといった更新に因る情報である。したがって、再訪問時に更新が起こっているとき、更新情報を把握しやすい形で提示することがユーザの情報収集の支援となる。

更新による情報の追加に関して、**RSS**（RDF Site Summary、または Really Simple Syndication, Rich Site Summary）[2] や **Atom** [3] などのフォーマットで発信される更新情報を捉え、その情報をユーザに提示する RSS リーダなどが徐々に実用化されてきている。しかし、それらの更新情報は Web サイト側によって記述、発信されるため、そこから得られる情報は Web ページの閲覧者が必要としている情報とは限らない。また、それらを発信していない Web サイトも多数ある。

更新による情報の削除に関して、削除が起こった際にユーザの情報収集を支援するインタフェースは見られない。したがって、Web ページへの再訪問時、更新情報などのユーザが必要とする情報をユーザに適した形で提示するインタフェースが求められている。

本研究の目的

本研究では、Web ページへの再訪問時の情報収集支援を目的として、ユーザが訪問している Web ページ内の情報を蓄積し、再訪問時に更新差分を提示するシステムを提案する。さらに、更新差分を Web ページの概観と併せてユーザに把握しやすく提示するインタフェースを提案する。提案したインタフェースを実現するシステムを試作し、考察を行う。

本論文の構成

本論文の構成について述べる。第 2 章では、Web ブラウジングにおける Web ページへの再訪問の現状と再訪問時の Web ページに存在する情報を考察し、従来のブラウジングインタフェースの問題点を述べる。第 3 章では、それらの問題点を解決するために Web ページへの再訪問時の情報収集を支援するインタフェースを提案する。第 4 章では、第 3 章の提案を基

に設計したシステム「**StackOverview**」の詳細について述べる．第 5 章では，StackOverview の活用例を述べ，従来のインタフェースと比較し，その利点を考察する．第 6 章では，本研究に関連する研究について述べ，比較と議論を行う．最後に，第 7 章で本論文をまとめる．

第2章 Web ブラウジングと再訪問の現状

2.1 再訪問

Andy Cockburn らは、Web ブラウザの「戻る」、「進む」、「履歴」、「ブックマーク」などを用いて、過去に閲覧した経験のある Web ページを再度閲覧することを**再訪問**と定義した [4]. ユーザが Web ページを閲覧した回数の総数を `total_visit_count`, ユーザが閲覧した URL の総数を `total_URL_count`, Web ページへの再訪問率を R とすると, R は式 2.1 で表わされる. 彼らの調査 [1] によって, Web ブラウジングにおいて R が 81% となることが明らかにした.

$$R = 100 \times \frac{\text{total_visit_count} - \text{total_URL_count}}{\text{total_visit_count}} \quad (2.1)$$

また, 彼らはユーザの再訪問を支援するために「戻る」、「履歴」、「ブックマーク」を統合したシステム「WebView」の開発を行った [5, 6]. その他の再訪問を支援するシステムとして, RSS, Atom などのフォーマットで発信されるフィードが挙げられる. このフィードを利用することによって Web ページの更新を効率よく捉えることができる.

2.2 再訪問時の情報収集

上記のように, 再訪問は Web ブラウジングにおいて重要な役割を持ち, その再訪問を行うためのシステムの研究も進められている. しかし, 再訪問した後の Web ページでの情報収集インタフェースについての考察は不十分である. 本研究では, この再訪問時の Web ページにおける情報収集インタフェースに焦点を当てる.

2.2.1 情報の分類

本研究では, 再訪問後の Web ページにある情報のうち特定の情報を 3 種類に分類して扱う. 以下にその分類の定義を述べる.

追加情報

Web ページの更新によって新規に追加された情報を「**追加情報**」と定義する。追加情報は、ウェブログやニュースサイトのトップページ、電子掲示板のスレッドなど更新頻度の高い Web ページでよく見られる。

注目情報

ユーザが Web ページを訪問したとき、再度閲覧する必要があると認識した情報を「**注目情報**」と定義する。注目情報は、Web 上の大部分のサイトに現れる。一般的には、前述のウェブログやニュースサイトの記事、プログラム言語のリファレンスなどが注目情報に当てはまる。ただし、再度閲覧する必要があると認識するユーザが存在するならば、どのような情報でも注目情報になり得る。

削除情報

ユーザに注目情報として認識されていたが、Web ページの更新によって削除された情報を「**削除情報**」と定義する。追加情報と同様に、ウェブログやニュースサイトのトップページなど更新頻度の高い Web ページでよく見られるが、電子掲示板のスレッドのような記事の削除が少ない Web ページでは余り見られない。

2.2.2 再訪問時の問題

更新による Web ページの変化

Web 上の情報は頻繁に更新されるが、再訪問時に Web ページが更新されていたとき、追加情報、削除情報の把握が非常に困難である。

ウェブログを例に挙げる。再訪問時のウェブログに追加情報が存在するかどうかを確かめるには、記事の日付や内容などを確認する必要がある。これは、存在するかどうか分からない情報を探すことになり非効率的である。また、ユーザが過去の訪問においてある記事を注目情報として認識していたとする。後日、再訪問を行ったとき、更新によってその記事がトップページから削除されていたり、あるいは記事自体が削除されていたりすることがある。

まず、追加情報の把握が困難な原因は、追加情報が既存の情報と混ざってしまうことである。次に、削除情報の把握が困難な原因は、情報自体が消えてしまうことである。したがって、削除情報を Web ページに再現し、さらに追加情報、削除情報を既存の情報と区別して提示することができればよい。

情報の位置認識とアクセス

再訪問を行った Web ページから過去に見た情報（注目情報）を探すとき，その情報の正確な位置認識とその情報へのアクセスが困難である．

Boungwon Suh らによると，Web ページの 69% が 1 画面以上の長さを持つ [7]．そのため，スクロールを行わなければ探している情報が視認できないことが多々ある．探している情報にたどり着くためには，徐々にスクロールしていく，または，目算によってスクロールバーを移動させる必要がある．しかし，再訪問する毎にこのような手段をとることは非効率である．

この問題は，探している注目情報と Web ページ全体の位置関係が示されていないことが原因である．情報の位置関係が示されていれば，注目情報の正確な位置認識，さらに，その情報へのアクセスが容易になる．したがって，探している注目情報と Web ページ全体の位置関係を提示するインタフェースを考える．これには，Web ページ全体の情報の概観を提示すればよい．現状，スクロールバーが Web ページ全体の概観を現していると言えるが，スクロールバーには現在見ている範囲の情報と Web ページ全体の位置関係のみが提示されている．つまり，スクロールバーが提示する情報に加えて，注目情報の位置を Web ページの概観に提示するインタフェースがあればよい．

第3章 Web ページの概観と更新差分

本研究では，前章で述べた問題点を解決し，Web ページへの再訪問時の情報収集を支援するために，以下の要素を備えたインタフェースを提案する．

- Web ページの概観を提示する．
- 削除情報を再現する．
- 注目情報 / 追加情報 / 再現された削除情報を既存の情報と区別して提示する．

3.1 概観提示

Web ページの概観として，Web ページ全体の縮小画像を一画面に収まる大きさを提示する（図 3.1 参照）．この概観によって，現在見ている範囲の情報だけでなく，スクロールしなければ見えない位置にある情報と Web ページ全体の位置関係をユーザが理解できるようにする．さらに，次節で詳細を述べるが，概観に注目情報 / 追加情報 / 削除情報へのリンクを設置する．そのリンクをクリックするだけで概観から元の Web ページの詳細に容易に移動することができるようにする．

3.2 情報の提示

3.2.1 追加情報 / 注目情報

Web ページへの訪問時に追加情報や注目情報が存在するとき，その情報に色分けによる強調を行う．また，概観上の対応する位置にその情報へのリンクを設置する．この手法により，情報が追加されたことの認識と情報の位置認識，その情報へのアクセスをユーザが容易に行えるようになる．

図 3.2 に追加情報 / 注目情報の提示手法を示す．図中の黄色で強調表示された矩形が注目情報である．図右において Web ページの詳細と概観上に追加された赤色で強調表示された矩形が追加情報である．

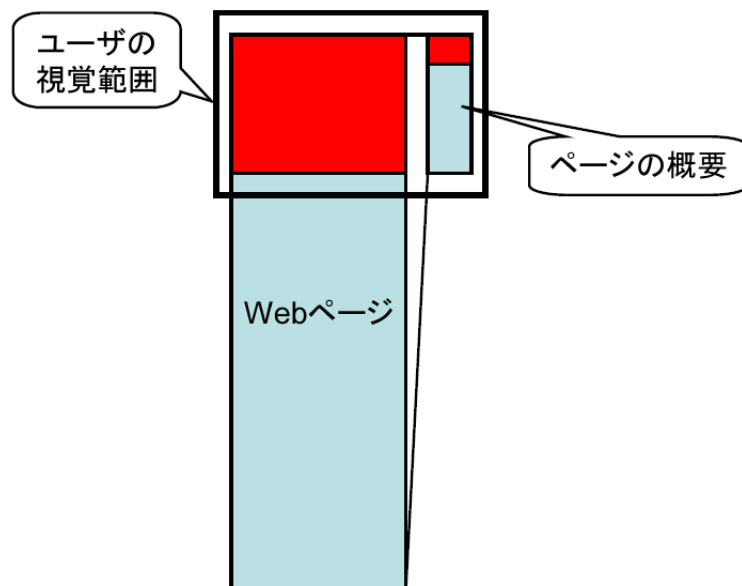


図 3.1: Web ページの概観

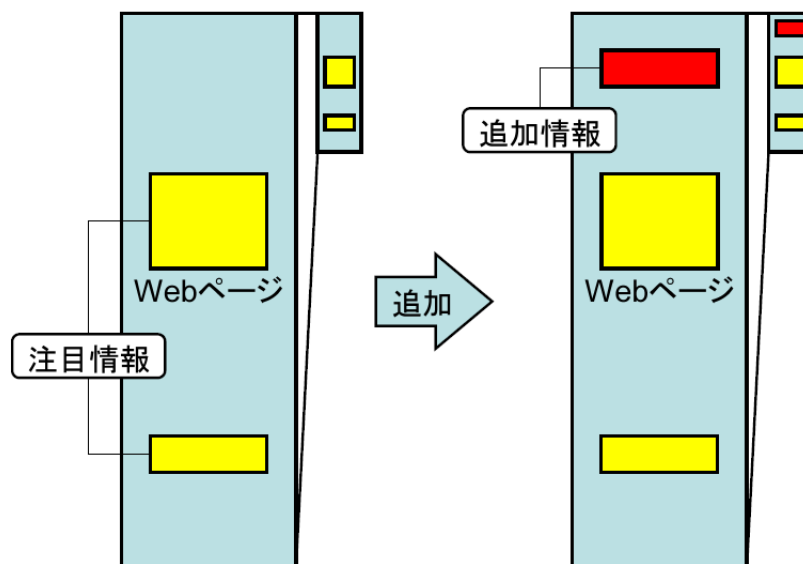


図 3.2: 追加情報 / 注目情報の提示

3.2.2 削除情報

Web ページへの訪問時に削除情報が存在するとき、その情報を Web ページの元あった位置に再現し、色分けによる強調表示を行う。ユーザは、依然その情報が必要であればそのままの状態を維持し、情報に価値が無くなったならばその情報を削除する。また、概観上の対応する位置にも削除情報へのリンクを設置する。この手法により、情報が削除されたことの認識とその情報の再現、情報の位置認識とその情報へのアクセスをユーザが容易に行えるようになる。

図 3.3 に削除情報の提示手法を示す。図中央で Web ページの詳細から削除された矩形が削除情報であり、図右において Web ページの詳細と概観上に青色に強調表示されて再現される。ただし、ここで再現されるのは削除情報へのリンクだけであり、詳細な内容は提示されない。次節で詳細を述べる。

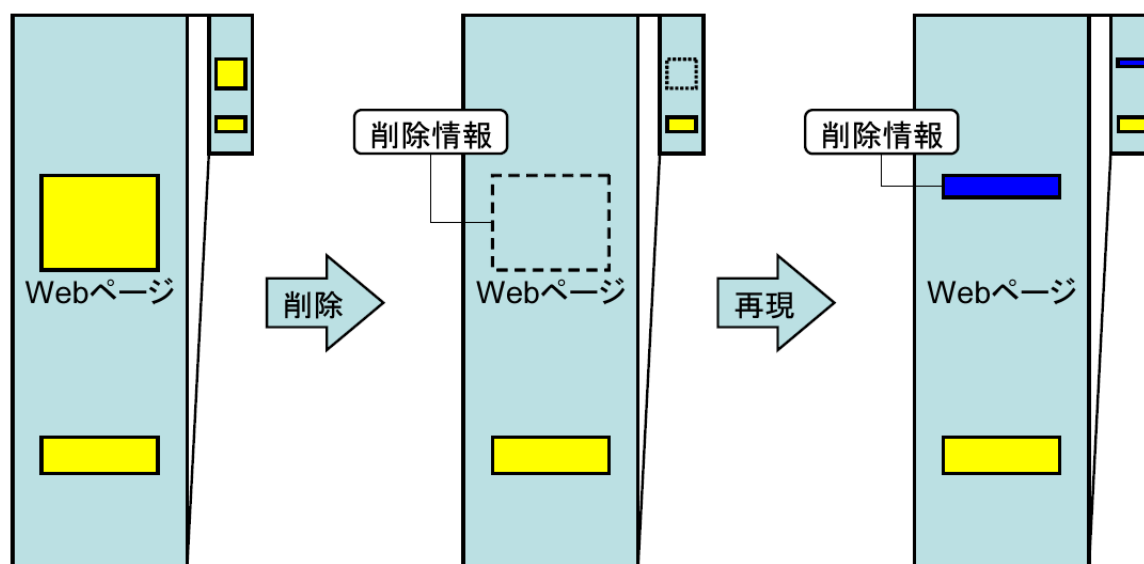


図 3.3: 削除情報の提示

削除情報 / リンクの展開

削除情報を再現するとき、以下の 2 点に注意しなければならない。第一に、再現される前の Web ページの背景をできる限り保持することである。そうすることで Web ページの変化が少なくなり、ユーザが混乱することなく削除情報再現後の Web ページを閲覧できる。第二に、ユーザの需要に応じて削除情報にすばやくアクセスできることである。この 2 点を踏まえて、削除情報はその情報を持つ文書へのリンクの形で元の Web ページに再現する。この

リンクをクリックすることでリンク先の文書が元の Web ページ内に展開され、ユーザが削除情報の詳細を閲覧できるようにする。なお、リンク先の展開によって元の文書は長大化する。そのため、そのままでは 2.2.2 節で述べた情報の位置認識とアクセスの問題が発生する。しかし、この問題は前節の概観提示により解決済みである。

図 3.4 に削除情報 / リンクの展開を示す。図左でリンクと削除情報をクリックすることにより、図右でそれらが展開される。展開後の Web ページ全体の縮小画像が一画面に収まる大ききで提示されるため、概観は一定の大きさを維持する。

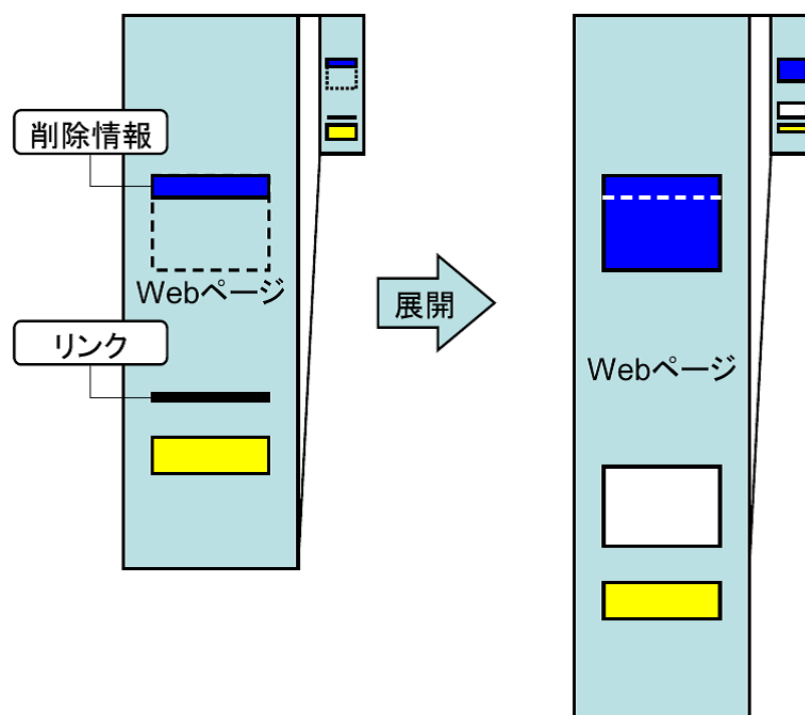


図 3.4: リンク / 削除情報の展開

第4章 StackOverviewの実装

前章で述べたインタフェースに基づいて設計を行ったシステム **StackOverview** の試作を行った。システムの開発環境、および実行環境は以下の通りである。

- OS: Microsoft Windows XP Home Edition Version 2002 Service Pack 2.
- 開発言語: JavaScript, XUL.
- 開発環境: Mozilla Firefox 1.5.0.1 日本語版 for Windows [8].

4.1 システムの概要

StackOverview は、Web ブラウザ **Firefox** の拡張機能（プラグイン）として実装を行った。Firefox は、Windows, Unix/Linux, Macintosh のクロスプラットフォームで同じインタフェースを提供するようにオープンソースで開発されている。ブラウザ自体の入手が容易であり、さらに、拡張機能の部分のみを .xpi ファイルとして配布できるため、幅広い環境でのユーザ評価を容易に行えるという利点がある。

拡張機能の内部では、XUL (XML User Interface Language) でユーザインタフェース（以下 UI）を記述する。XUL とは、インターネットからの接続で動作可能な機能に富んだ、クロスプラットフォームのアプリケーションの作成を可能にする Mozilla の XML ベースの UI 言語である。XUL で記述された UI および Web ページは DOM (Document Object Model) を持っている。この DOM を JavaScript を用いた DynamicHTML で操作することにより、UI および Web ページとのインタラクションを行う。

4.2 概観提示

4.2.1 縮小画像

StackOverview は、Web ページの右側にそのページ全体の縮小画像を用いた概観を提示する（図 4.1 参照）。



図 4.1: Web ページの概観

縮小画像は、HTML の canvas 要素を UI に埋め込むことによって表示される。canvas 要素は、WHATWG (Web Hypertext Application Technology Working Group) [9] の Web Application 1.0 の一部として標準化中のカスタム描画を行うための要素であり、現在 2 次元グラフィック機能を提供している。この canvas 要素を用いることによって、閲覧している Web ページと全く同じ表示が得られる。また、マウスポインタが概観上にある場合は 1 秒毎、それ以外の場合は 2.5 秒毎に縮小画像を更新する。そのため、常に最新の Web ページの縮小画像が表示される。

4.2.2 スクロール機能

提示された概観は、スクロール機能を提供する。すなわち、概観上でマウスを操作することによって、Web ページの詳細のスクロールを可能にする。Web ページの右側への概観の配置とスクロール機能によって、ユーザは通常 Web ページに表示されるスクロールバーと同様の感覚で StackOverview を使用できる。一般的なスクロール機能の例として、Internet Explorer などの Microsoft Windows XP の標準機能が持つスクロールバーに実装されている機能を表 4.1

に示す。また、StackOverview のスクロール機能を表 4.2 に示す。

表 4.1: 一般的なスクロールバーの機能

機能名称	動作
ここへスクロール	スクロールバー上のマウスポインターのある位置までスクロールする
上	先頭位置に移動する
下	最後尾に移動する
1 画面分上へスクロール	1 画面分上へスクロールする
1 画面分下へスクロール	1 画面分下へスクロールする
上へスクロール	約 2 行分上へスクロール
下へスクロール	約 2 行分下へスクロール

表 4.2: StackOverview のスクロール機能

操作	動作
概観上でクリック	クリックした方向に少量スクロール
概観上でダブルクリック	ダブルクリックした位置までスクロール
概観上でドラッグ	カーソルに合わせてスクロール

4.3 情報の蓄積

4.3.1 Web ページの自動収集

StackOverview は、ユーザが Web ページを訪問するとき、その Web ページの文書を Firefox の「履歴」や「キャッシュ」とは別に URL 毎に**自動収集**する。

4.3.2 注目情報のスクラップ

ユーザが閲覧している Web ページに注目情報が存在するとき、ユーザはその範囲を選択し、コンテキストメニューの選択を行うことによりその注目情報のスクラップを行う。図 4.2 に選択範囲に対するコンテキストメニューからのスクラップを示す。

スクラップでは、まず選択範囲の HTML オブジェクトを抽出する。抽出したオブジェクトは XML 形式で扱えるノードとして扱う。スクラップは URL 毎に保存する。過去にその Web ページのスクラップがない場合は抽出されたオブジェクトのノードを持つ XML ファイルを作成し、既にその Web ページのスクラップに対応する XML ファイルが存在する場合はその

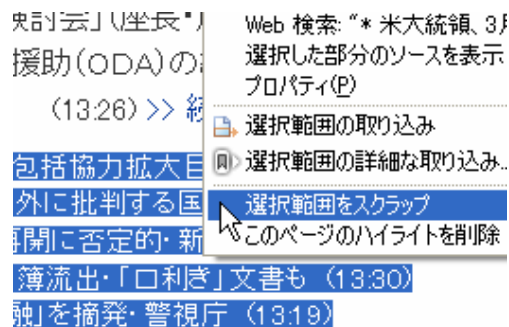


図 4.2: コンテキストメニューからのスクラップ

XML ファイルにノードを追加する．作成する XML ファイルは以下のようなになる．

```
<?xml version="1.0" encoding="Shift_JIS"?>
<content>
  <scrap>                                     //スクラップ
    <div>.....</div>
  </scrap>
  <scrap>                                     //スクラップ
    <table>.....</table>
  </scrap>

  .....

  <scrap>                                     //スクラップ
    <a href="...">.....</a>
  </scrap>
</content>
```

4.4 情報の抽出

4.4.1 追加情報

ユーザが Web ページを再訪問したとき，前回の訪問時に保存しておいた文書と現在の Web ページの文書を比較し，差分を取る．比較の結果，Web ページに追加されている部分があったとき，その部分の HTML オブジェクトを追加情報とする．

4.4.2 注目情報 / 削除情報

ユーザが Web ページを再訪問したとき、その Web ページのスクラップが存在するならば、スクラップと現在の Web ページの文書を比較する。まず、XML ファイルに保存されているスクラップを現在の Web ページから検索する。検索の結果、そのスクラップが現在も Web ページに存在すれば、そのスクラップに対応する Web ページ内の HTML オブジェクトを注目情報とする。現在の Web ページに存在しなければ、そのスクラップのノードを削除情報とする。

4.5 情報の提示

前節において抽出された注目情報 / 追加情報 / 削除情報を、Web ページの詳細と概観に強調して提示する。

4.5.1 注目情報

注目情報が存在するとき、Web ページの詳細において注目情報のオブジェクトのスタイルシートを変更し、backgroundColor 属性の値を #ffff00 とする。また、概観を構築する DOM ツリーに注目情報を示すオブジェクトを挿入する。このオブジェクトをクリックすることで Web ページの詳細において注目情報を閲覧することができる。これらの操作によって注目情報が Web ページの詳細と概観の両方において「黄色」で強調表示される（図 4.3 参照）。

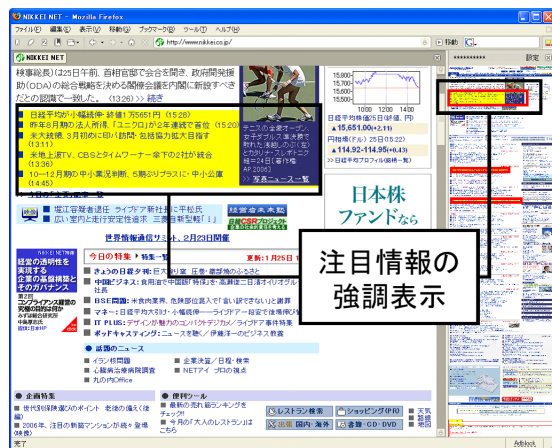


図 4.3: 注目情報の提示

4.5.2 追加情報

追加情報が存在するとき、追加情報の場合と同様に、スタイルシートの変更と概観の DOM ツリーへのオブジェクトの挿入を行う。ただし、スタイルシートの backgroundcolor 属性の値は #ff0000 とする。これらの操作によって追加情報が Web ページの詳細と概観の両方において「赤色」で強調表示される（図 4.4 参照）。



図 4.4: 追加情報の提示

4.5.3 削除情報

削除情報が存在するとき、Web ページの構成する DOM ツリーの削除情報が過去に存在した位置に削除情報を再現したオブジェクトを挿入する。このオブジェクトは、削除情報の内容を示す見出しを持ったリンクのみを表示する。さらに、他の情報の場合と同様に、再現したオブジェクトのスタイルシートの変更と概観の DOM ツリーへのオブジェクトの挿入を行う。ただし、スタイルシートの backgroundColor 属性の値は #0000ff とする。これらの操作によって、削除情報が Web ページの詳細と概観の両方において「青色」で強調表示される（図 4.5 参照）。また、表 4.3 に強調表示される情報とその色の対応を示す。

表 4.3: 強調される情報と色

強調される情報	色	カラーコード
注目情報	黄	#ffff00
追加情報	赤	#ff0000
削除情報	青	#0000ff



図 4.5: 削除情報の提示

4.5.4 inlineLink によるリンクの展開

三浦らによる我々の先行研究である「**inlineLink**」[10, 11]を用いて、削除情報 / リンクの展開を行う。挿入された削除情報を示すオブジェクトや Web ページに元々存在するリンクアンカーのリンク先の文書を Web ページ内に展開する。図 4.6 に inlineLink によるリンクの展開を示す。

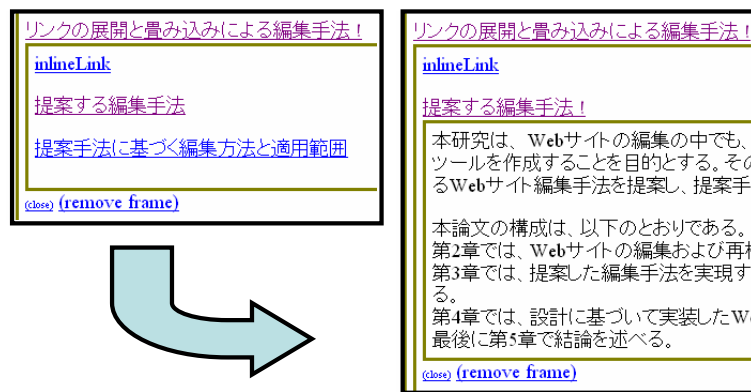


図 4.6: inlineLink によるリンクの展開

第5章 システムの活用例

5.1 ウェブログの記事

ユーザが Web 上で過去に見た情報を探す状況を想定する。その情報を見た Web ページをブックマークなどを利用して再訪問しても、その情報の Web ページ内における正確な位置を記憶しておくことは難しい。このような状況において、従来のインタフェースを用いた場合と StackOverview を用いて過去の訪問時にその情報を注目情報としてスクラップとして保存しておいた場合を比較する。

探す情報をウェブログの記事とする。図 5.1 が従来のインタフェース、図 5.2 が StackOverview によるウェブログへの再訪問を示す。従来のインタフェースの場合、Web ページを訪れた時点では探している情報の位置が不明瞭なため、大まかな位置を目指して画面をスクロールさせる必要がある。一方、StackOverview の場合、概観に注目情報の位置が明確に示されており、また、概観をダブルクリックすることで情報の位置までスクロールできるため、探している情報に即座に得ることができる。

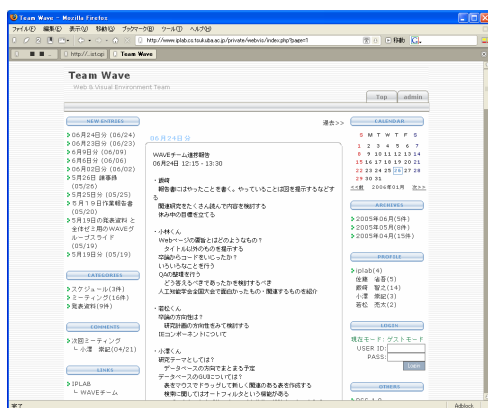


図 5.1: 従来のブログ再訪問

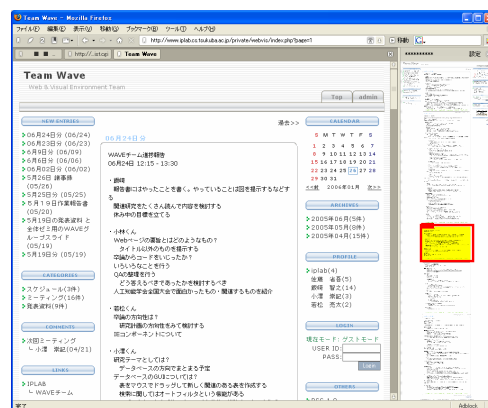


図 5.2: StackOverview によるブログ再訪問

5.2 ニュースサイトの記事

電子掲示板やニュースサイトは、頻繁に更新が行われる種類の Web サイトである。新しい情報が逐次追加され、古くなった情報はトップページから削除されてしまうため、必要な情報を効率よく閲覧する必要がある。

ニュースサイトへの再訪問時における StackOverview の活用例を述べる。図 5.3 が従来のインタフェース、図 5.4 が StackOverview によるニュースサイトへの再訪問を示す。従来のインタフェースの場合、「新しい情報はページ上部に多い」といった大まかな指針を基に Web ページを閲覧するしかなかった。一方、StackOverview の場合、概観の上部に追加情報の位置、概観の下部に削除情報の位置がそれぞれ明確に示されており、更新による変化を容易に把握できる。

したがって、Web ページの上部にある最新の追加情報を優先的に閲覧してから他の情報を閲覧したり、あるニュースの一連の記事に対して、削除情報になった過去の記事から新しい記事へと時系列に沿って閲覧したりできるようになる。



図 5.3: 従来のニュースサイト再訪問



図 5.4: StackOverview によるニュースサイト再訪問

第6章 関連研究

Saul Greenberg らによって Web ブラウジングにおける Web ページへの再訪問についての研究が行われている [4, 1]. 彼らは, Web ブラウジングにおけるユーザの行動を調査し, その中で再訪問の重要性を述べている. 彼らの研究では, 再訪問を行うためのインタフェース「**Web View**」などを考案している [5, 6]. 一方, 本研究では Web ページへの再訪問を行った後の情報収集を支援するインタフェースの提案と試作を行った.

Web ページの概観提示に関する研究には, Bongwon Suh らによる「**Popout Prism**」[12, 7] と Patrick Baudish らによる「**Fishnet**」[13] がある. これら 2 つの研究では, 本研究と同様に Web ページ全体の縮小画像を用いて概観を提示している. さらに Fishnet では, 魚眼レンズのような視覚効果によって Web ページの上端や下端の一部を縮小することにより Web ページ概観を提示する方法も用いている. これらの研究では, Web ページ内での単語検索の結果を Web ページの詳細と概観上に強調表示するというを行っているが, 本研究ではこれまでに述べてきた再訪問において有用な情報を強調表示し, さらに, それらの情報とのインタラクションのためのインタフェースとして概観を用いる.

また, 本研究内で用いた inlineLink [10, 11] に関して, リンク先文書の展開に関する研究に Polle T. Zellweger らによる「**Fluid Open Hypermedia**」[14, 15] がある. Fluid Open Hypermedia では, リンク先の文書の展開や情報への注釈情報の追加を Web ページの背景を保ったまま行う.

本研究では, Web ページ訪問時にユーザが注目した情報をスクラップし, その更新の追跡を行った. 杉浦, 古関らの「**Internet Scrapbook**」[16] では, Web ページにおいてユーザが注目した情報のスクラップを行い, 注目情報を集めた仮想的な Web ページを作成することによりそれらの情報の更新を追跡した. 仮想的なページに情報を集めることで必要な情報だけの提示を実現したが, 本研究では強調表示を用いて情報の Web ページ内での位置を明確にすることによって背景を保ったまま必要な情報を見ることができる.

第7章 結論

本論文ではまず、Web ページへの再訪問と再訪問時の情報収集の現状を考察した。その際、Web ページの情報を注目情報 / 追加情報 / 削除情報の 3 種類に分類した。また、Web ページの更新による変化と情報の位置認識の問題について説明した。

それらの問題を解決するために、Web ページの更新差分と概観を用いてユーザに分かりやすく情報を提示し、再訪問時の情報収集を支援するインタフェースの提案を行った。さらに、提案したインタフェースを実現するシステム「StackOverview」を Mozilla Firefox の拡張機能として試作し、その活用例を述べた。

今後の展望としては、概観を用いた情報の提示手法と操作体系の改善、蓄積するデータのためのビューアの実装、ユーザ評価などを行い、使いやすいシステムの開発を進める予定である。

謝辞

本論文の執筆にあたり，田中二郎先生をはじめ，志築文太郎先生，三末和男先生，高橋伸先生には丁寧なご指導と適切なご助言を頂きました．ここに深く感謝申し上げます．また，田中研究室のメンバーの皆様にも大変お世話になりました．とりわけ WAVE チームの皆様には研究の全般にわたりご意見やご指導を頂きました．この場を借りてお礼申し上げます．

関連図書

- [1] Andy Cockburn, Saul Greenberg, Steve Jones, Bruce McKenzie, and Michael Moyle. Improving web page revisitation: Analysis, design and evaluation. *IT & Society*, Vol. 1, Issue 3, Winter 2003, pp. 159–183, 2003.
- [2] W3C. <http://www.w3.org/>.
- [3] Atom Wiki. <http://www.intertwingly.net/wiki/pie/FrontPage>.
- [4] Linda Tauscher and Saul Greenberg. Revisitation patterns in world wide web navigation. In *CHI '97*, pp. 399–406, 1997.
- [5] Andy Cockburn, Saul Greenberg, Bruce McKenzie, Michael Jasonsmith, and Shaun Kaasten. Webview: A graphical aid for revisiting web pages. In *OZCHI '99*, pp. 15–22, 1999.
- [6] Shaun Kaasten and Saul Greenberg. Integrating back, history and bookmarks in web browsers. In *CHI '01*, pp. 379–380, 2001.
- [7] Bongwon Suh, Allison Woodruff, Ruth Rosenholtz, and Alyssa Glass. Popout prism: Adding perceptual principles to overview+detail document interfaces. In *CHI '02*, pp. 251–258, 2002.
- [8] Mozilla Japan. <http://www.mozilla-japan.org/>.
- [9] WHATWG. <http://www.whatwg.org/>.
- [10] Motoki Miura, Buntarou Shizuki, and Jiro Tanaka. inlinelink: Realization of inline expansion link methods on a conventional web browser. In *APCHI2004*, pp. 272–282, 2004.
- [11] 小野田紗智, 三浦元喜, 田中二郎. 展開と畳み込みに基づく web 文書の構造編集ツール「javaara」. 日本ソフトウェア科学会第 21 回大会, 2003.
- [12] Allison Woodruff, Andrew Faulring, Ruth Rosenholtz, Julie Morrision, and Peter Pirolli. Using thumbnails to search the web. In *CHI '01*, pp. 198–205, 2001.
- [13] Patrick Baudisch, Bongshin Lee, and Libby Hanna. Fishnet, a fisheye web browser with search tern popouts: a comparative evaluation with overview and linear view. In *AVI '04*, pp. 133–140, 2004.

- [14] Niels Olof Bouvin, Polle T. Zellweger, Kaj Grønæk, and Jock D. Mackinlay. Fluid annotations through open hypermedia: using and extending emerging web standards. In *WWW '02*, pp. 160–171, 2002.
- [15] Polle T. Zellweger, Bay Wei Chang, and Jock D. Mackinlay. Fluid links for informed and incremental hypertext browsing. In *CHI '99*, pp. 7–8, 1999.
- [16] Atsushi Sugiura and Yoshiyuki Koseki. Internet scrapbook: Automating web browsing tasks by demonstration. In *UIST '98*, pp. 9–18, 1998.