



平成 18 年度 修士論文

携帯電話向け

Web クリッピングシステムの設計と実装

電気通信大学 大学院情報システム学研究科
情報システム設計学専攻

0550004 五十嵐 正浩

指導教員 多田 好克 教授
 田野 俊一 教授
 植野 真臣 助教授

提出日 平成 19 年 1 月 30 日

目次

第 1 章	はじめに	6
第 2 章	背景と目的	9
2.1	背景	9
2.2	目的	12
2.3	利用シーン	13
第 3 章	関連研究	17
3.1	コンテンツの変換に関する研究	18
3.2	Web ページの分割提示に関する研究	20
3.3	Web ページの保存・編集に関するツール	21
3.3.1	ScrapBook	21
3.3.2	紙 copi	22
第 4 章	設計	23
4.1	システム設計の前提条件	23
4.2	Web クリッピングとは	23
4.3	システムの概要	24
4.4	設計方針	24
4.5	ブロック化対象タグの決定	27
第 5 章	実装	28
5.1	実装概要	28
5.2	拡張機能	29

5.2.1	ブロック表示機構.....	29
5.2.2	ブロック選択機構.....	33
5.2.3	ブロックサイズ変更機構.....	35
5.2.4	切り抜きファイル作成機構.....	39
5.3	投稿・閲覧サイト.....	42
第6章	評価.....	43
6.1	評価方法.....	43
6.2	実験結果.....	44
6.2.1	ニュースサイトからの切り抜き.....	44
6.2.2	表を含む Web ページの切り抜き.....	47
6.3	考察.....	49
6.3.1	本システムで作成した Web ページの有効性.....	49
6.3.2	携帯電話に対応しているマークアップ言語.....	50
第7章	問題点と課題.....	52
7.1	切り抜き精度に関する問題.....	52
7.2	初期表示のブロックサイズについて.....	53
7.3	フレームを用いた Web ページへの対応.....	54
7.4	機能の拡張性.....	55
7.5	投稿・閲覧サイトの改善.....	56
7.6	切り取った Web ページの融合.....	56
第8章	おわりに.....	57
付録A	本システムの操作方法.....	63
付録B	著作権について.....	69

図目次

図 2.1: 携帯電話でも表示できる Web ページの例	10
図 2.2: 携帯電話で表示できない Web ページの例	11
図 2.3: 本システムを使用しない場合の例	15
図 2.4: 本システムを使用した場合の例	16
図 3.1: 携帯インターネットの普及率	18
図 4.1: システム概要	25
図 5.1: HTML ソースと DOM ツリー	30
図 5.2: HTML ソースと DOM ツリーの対応	30
図 5.3: CSS の命令構造	31
図 5.4: wec_flag の値の設定	34
図 5.5: ブロックサイズのレベル設定例	36
図 5.6: wec_level1 の例	36
図 5.7: wec_level3 の例	36
図 5.8: ブロックサイズの一括変更の例	37
図 5.9: ブロックサイズの局所変更の例	38
図 5.10: オリジナルツリーの生成	40
図 5.11: 作成したファイルの保存先例	42
図 6.1: 切り抜く元の Web ページ	44
図 6.2: 図 6.1 の Web ページをそのまま携帯電話で表示した画面	45
図 6.3: 図 6.2 で切り抜き範囲に該当するコンテンツを表示した画面	46
図 6.4: WeC を用いて作成した Web ページを表示したときの画面	46

図 7.1: Web ページ中のテキストとそれに対応するツリー構造	53
図 7.2: 初期表示のレベルによるブロックサイズの変更例.....	54
図 7.3: フレームを用いた Web ページの DOM ツリーの例	55

表目次

表 3.1: コンテンツ変換方式と本システムの機能の違い.....	19
表 3.2: 分割提示方式と本システムの機能の違い.....	21
表 3.3: 関連ツールと本システムの機能の違い.....	22
表 6.1: 各キャリアの端末の対応言語とその特徴.....	50

第 1 章

はじめに

近年、携帯電話の普及率増加が進み、1人1台持つ時代になってきた。さらに、端末の小型化や通信技術の発展により、携帯電話での Web ブラウジングがいたる所で行われるようになってきた。また、ディスプレイの高精細化や通信速度の高速化により、PCでの閲覧を想定して作成された Web ページを携帯電話でも同じように閲覧することができるブラウザも登場してきている。しかし、携帯電話などの画面サイズの小さい端末では、そういった PC の大画面を想定して作成された Web ページの閲覧は困難である。このように、携帯電話での Web ブラウジングには、画面サイズの問題と閲覧可能なコンテンツの最大サイズなど、様々な制限が存在する。

また、PC や携帯電話の普及により、インターネットの環境は身近なものになっている。それに伴い、旅行や行楽に出かける際に、事前に目的地や関連情報をインターネットで調べる機会も増えてきている。しかし、調べた結果は印刷して持っていくことが多く、情報量が多くなると大変扱い辛いものになってしまう。

携帯電話の普及に伴い、それに関する様々な研究・製品が発表されている。携帯電話における Web ブラウジングに関する既存の研究やブラウザでは、Web ページをいくつかの領域に分割し、その領域を表示させる手法が主流である。これは、閲覧する Web ページの構造が分かっている場合は、さほど問題なく目

的のコンテンツを閲覧することができるが、そうでない場合は困難なものになる。また、インターネットで調べた Web ページを印刷せずに携帯電話でその都度閲覧する方法も考えられるが、携帯電話の小さな画面に表示した Web ページの中から目的のコンテンツを再度探すのは手間がかかってしまう。

本研究では、PC で閲覧した Web ページの情報を持ち出す、ということに着目する。紙に印刷して持ち出す場合は、情報量が多くなると扱い辛いものになってしまうという問題が発生する。そのため、普及率の高い携帯電話に持ち出す方法を考えた。しかし、この場合にも画面サイズの小ささから発生する問題などがあるため、Web ページをそのまま表示させることは、ユーザの立場からすると好ましくない。

ここで、新聞や雑誌で必要な記事を切り抜いて貼り付けるスクラップブックに着目した。この関係を Web ページとコンテンツ、携帯電話に置き換えると、上記の問題が解決できると考えた。そこで本研究では、Web ページから必要なコンテンツを切り抜き、それを携帯電話で表示させるシステムを構築した。

本研究の主な目的は、Web ページの切り抜きと携帯電話での表示なので、携帯電話での一般的な Web ブラウジングは対象外とした。

本研究では、コンテンツの切り抜きの際に、Web ページを HTML タグによってブロック化し、それを選択することで切り抜く範囲を指定する。このブロックはサイズの動的な変更が可能なので、ユーザは必要なコンテンツのみを切り抜くことができる。切り抜き作業が終了したところで、切り抜いたコンテンツから構成される HTML ファイルが作成される。ユーザは、この HTML ファイルを本研究が提供する投稿・閲覧サイトに投稿し、このサイトに携帯電話からアクセスし、投稿した HTML ファイルを閲覧することで、作成した Web ページを閲覧することができる。

これにより、ユーザは煩雑なデータ管理から開放され、Web ページの持つリンクなどの機能を利用することができるようになる。

本論文は、次のような構成となっている。第2章では、インターネットと携帯電話の普及について述べ、次に、本研究の目的について述べる。第3章では、本研究の関連研究について述べる。第4章では、本研究の提案システムの設計について、そして第5章でシステムの実装について議論する。第6章では実装したシステムについての評価をし、考察を述べる。第7章で本システムに残された問題点と課題、今後の展望を示し、最後に第8章で本研究についてまとめる。

第 2 章

背景と目的

2.1 背景

近年、携帯電話の普及率が増加し、我が国の携帯電話の保有率は 90% にまで及び、1 人 1 台保有する時代になってきた[1][2]。さらに、端末の小型化やモバイル通信技術の進歩により、携帯電話での Web ブラウジングが一般的に行われるようになってきた。また、ディスプレイの高精細化や通信速度の高速化により、PC での閲覧を想定して作成された Web ページを携帯電話でも表示することができるブラウザも登場してきており、これらはフルブラウザと呼ばれる[3][4]。さらに、電気通信事業者では、さらなる通信速度の高速化と大容量データの送受信を可能にする研究を行っているため、今後ますます携帯電話での Web ブラウジングは普及していくと考えられる[5]。

しかし、携帯電話などの画面サイズの小さい端末では、PC での大画面を想定して作成された Web ページの閲覧は困難である。図 2.1 と図 2.2 にその例を示す。どちらも、PC と携帯電話で同じ Web ページを閲覧したときの違いを示している。図 2.1 は携帯電話でも Web ページが表示できる場合の例を示したもののだが、PC で閲覧している場合の左の画像と比べると、ページの概観が分からず、目的のコンテンツを探すのに手間がかかってしまう。次に、図 2.2 に示している例は、携帯電話では表示できない場合のものである。



a) 携帯電話での表示



b) PC での表示

図 2.1: 携帯電話でも表示できる Web ページの例



a) 携帯電話での表示



b) PC での表示

図 2.2: 携帯電話で表示できない Web ページの例

この原因は、電気通信事業者や端末に依存するもので、端末にダウンロードできる容量に制限がかけられているために起こる。それにより、閲覧できるコンテンツの最大サイズにも制限がある。この容量の制限に関する問題は通信速度の高速化により徐々に解消されてきている。

また、PC や携帯電話の普及などにより、インターネットの環境は身近なものになっている。2006年2月時点で日本のインターネット世帯浸透率は85.4%にまで達する[6]。それに伴い、旅行やレジャーに出かける際に、事前に目的地や関連する情報をインターネットで調べる機会も増えてきている。しかし、そういった調べた結果というのは、紙に印刷して持っていくことが多く、情報量が多くなると、それらは持ち運びや検索といった扱いが難しくなる。さらに、紙に印刷しているため、そこに書かれているリンク先の参照などは当然のことながらできない。しかし、現状ではPCで調べたWebページというのは、紙に印刷して外に持ち出すのが主流なままである。

2.2 目的

本研究は、PCで閲覧しているWebページからユーザが必要とする情報を切り抜き、それを携帯電話で表示・保存・活用可能とするシステムの提案、構築を行う。前述の背景を踏まえ、本システムを用いることによって、PCで閲覧しているWebページを印刷することなく、必要な情報の持ち運びを可能とすることが目的である。本研究の提案するシステムは、切り抜きの際に文章構造を示すHTMLタグ(以下:タグ)によってWebページをブロック化し、そのブロックを選択することでWebページの切り抜きを可能にする。また、紙に印刷するのと違い、Webページの形式で持ち出すので、そこに記載されたリンク先の参照も可能である。

2.3 利用シーン

この節では、本システムを実際に生活の中のどのようなシーンで利用するのか、以下に例を示す。

A 君は京都に旅行に行くことにした。現地で寺巡りをしようと思い、京都の寺に関する情報と、その周辺情報などをインターネットで調べてみた。すると、いくつかリストアップできたので、調べた Web ページを印刷してみた。印刷してみると、リストアップした量が多かったのか枚数が多くなってしまったのだが、A 君には調べた Web ページの情報を持ち出す手段が他に浮かばなかったのので、印刷した用紙の束を携えて旅に出た。

旅先で事前に調べた情報を見ようとしたのだが、印刷した枚数が多いため、中々見たい情報が印刷された用紙を見つけることができない。また、出発前の段階では必要ないと思い、寺へのアクセス方法が記載された Web ページを印刷しなかったのだが、現地に行ってみると道に迷ってしまった。その寺の情報について書かれた Web ページは印刷して持ってきており、同じサイトの別ページにアクセス方法が記載されていることは分かっているのだが、印刷物ではリンク先を参照することができない。そこで、携帯電話のブラウザでこのサイトにアクセスしてみたのだが、携帯電話は PC のディスプレイのような大画面に比べ、画面サイズが大変小さくなってしまいうので、PC 向けに作成された Web ページから目的のコンテンツを探すのは、とても困難なものになってしまう（図 2.3）。

そこで、本システムを使用していればどうなったか。まず、情報の管理については、リストアップした件数がいくら増えようと増えるのは携帯電話内のデータなので、印刷物の枚数が多くなり、管理が煩雑になることもない。さらに、端末に保存したデータの名称をユーザが分かりやすいものに変更しておけば、目的の情報を探すのも容易である。また、印刷物ではできなかったリンク先の

参照も、本システムを使用して切り抜いたものであれば、元の Web ページの HTML を編集して新たな HTML を作成しており、可能である（図 2.4）。

このように、本システムを用いることにより、煩雑なデータ管理が携帯電話でできるようになり、さらに HTML 情報も保持して切り抜くため、リンクなどの機能も利用することができる。

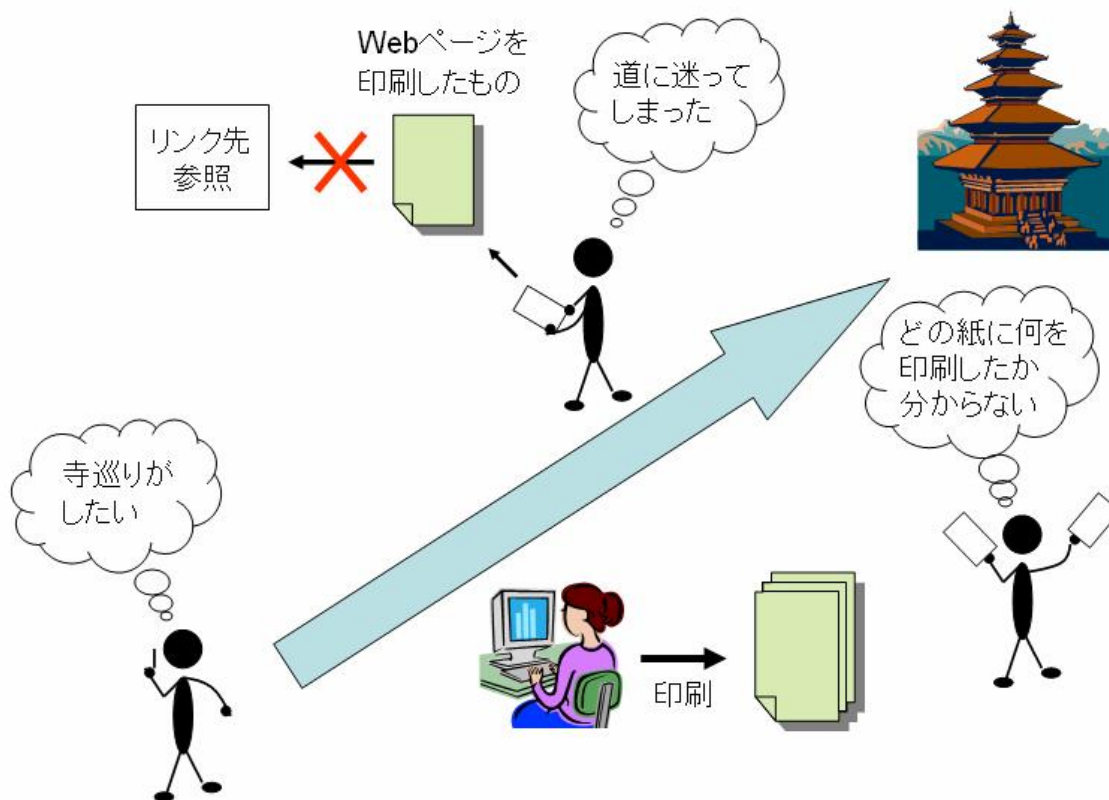


図 2.3: 本システムを使用しない場合の例

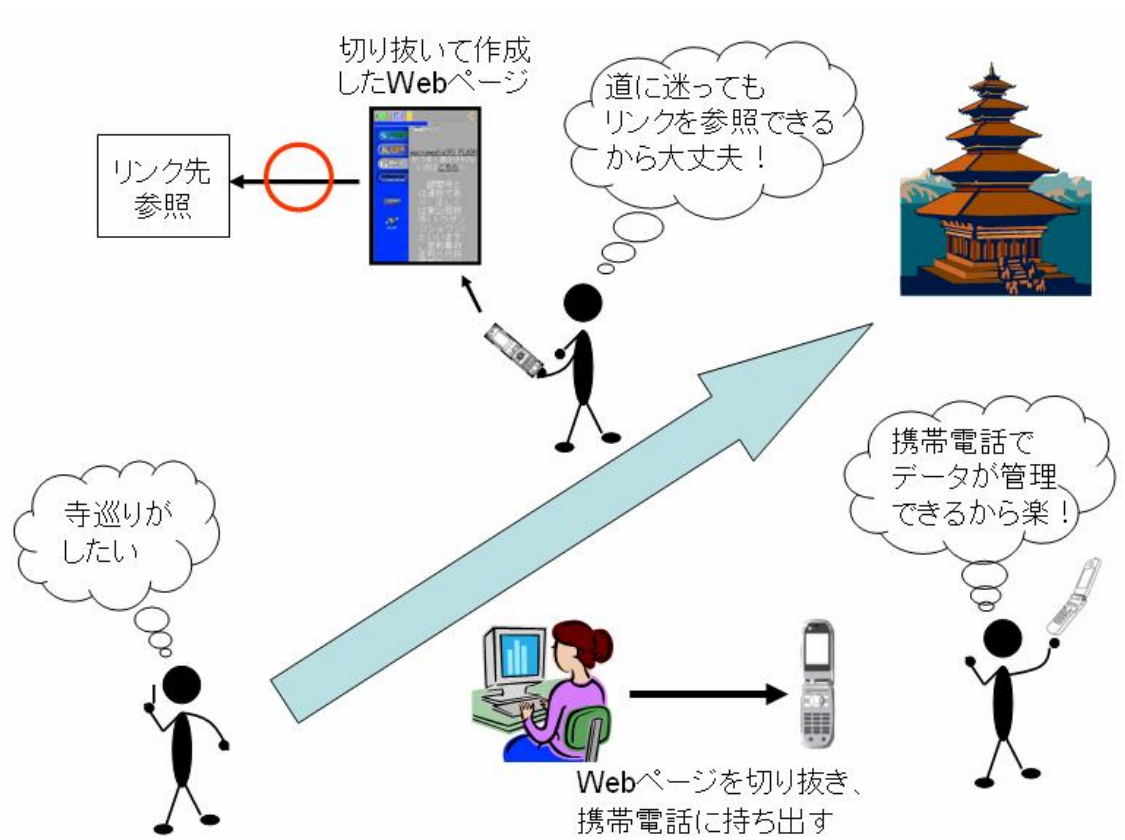


図 2.4: 本システムを使用した場合の例

第 3 章

関連研究

本章では、本研究と関連する既存の研究やツールについて述べる。

近年、携帯電話は急速に普及してきており、2006 年 3 月時点での携帯電話の普及率は、85.3%である（ただし単身世帯を含まない）[7]。また、日本のインターネット人口は 2006 年 2 月時点で 7,361 万 9 千人になり、今後も増える見通しである[6]。そうしたインターネット利用者の使用機器を見てみると、携帯電話・PHS 及び携帯情報端末が利用者の 81.2%と 8 割を超える。さらに、携帯インターネットの普及率は図 3.1 のように年々増えていることから、今後ますます携帯電話での Web ブラウジングは増えると予想される。しかし、携帯電話などの携帯情報端末には、画面の小ささや閲覧可能なコンテンツの最大サイズなど、様々な制限が存在する。このような制限の中で情報量の多いリッチなコンテンツを閲覧するためには、以下の方法が考えられる。

- リッチなコンテンツを携帯電話で閲覧可能なコンテンツに変換する
- 最初に Web ページの概観を提示し、Web ページを区分けした領域を選択することで、目的のコンテンツを探しやすくする

そこで以下では、それらの研究について詳しく述べる。また、本研究と同様に Web ページから切り抜きを行う既存のツールについても言及していく。

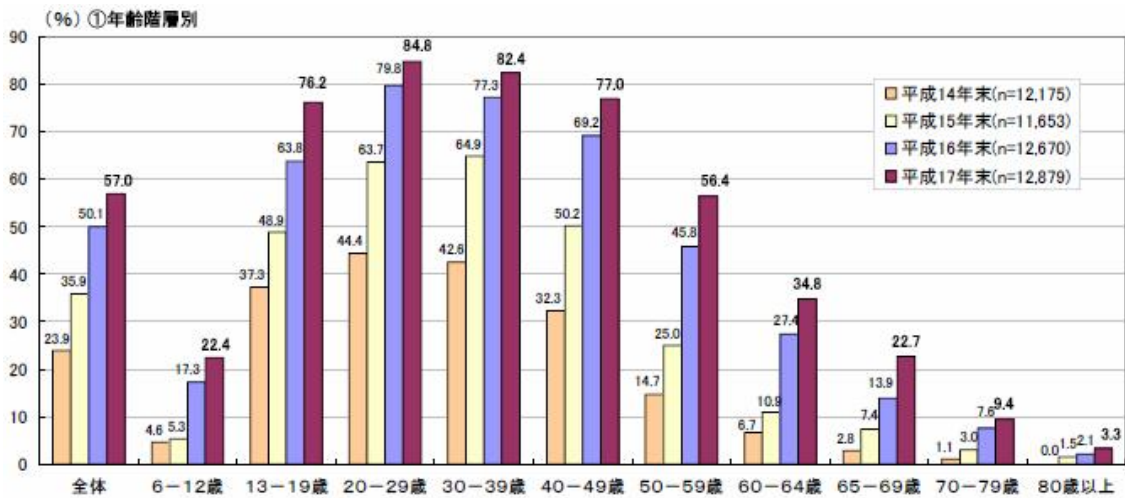


図 3.1: 携帯インターネットの普及率 (参考文献[2]より引用)

3.1 コンテンツの変換に関する研究

携帯電話には表示できない情報量の多いコンテンツを携帯電話上に表示することを目的とした従来研究では、閲覧するコンテンツの簡略化やコンテンツのインデックス作成などによるコンテンツの閲覧を実現しているものがある。

C.R.Anderson ら [8]は、PC での閲覧向けに作成された Web ページを、ユーザのプロファイル情報を用いて個人化することで、携帯電話に表示可能な小さな Web ページを作成している。具体的には、まず、ユーザの Web ブラウジング履歴から閲覧したページに現れる単語を解析し、プロファイル情報を作成する。そして、Web ページからユーザのプロファイル情報に合わない部分を省略する。

M.Jones ら [9][10][11]は、PC での閲覧向けに作成された Web サイトのインデックスページにあるリンクを解析することで、文字列のみからなるサイトのツリー型メニューを作成している。そしてユーザは、そのメニューを用いることにより、目的とするページを容易に探し出すことができる。これと同様のものに、Web サイトのメニューを作成する研究に O.Buyukkokten ら [12][13]が

ある。これは、PC 向けに作成された Web サイトのリンクから、サイト全体のインデックスを作成する。インデックスは、サイト内のページタイトルを表しており、ユーザはそのインデックスを用いることにより、サイト全体の構成を簡単に把握することができる。さらに、そのインデックス内でページタイトルを選択すると、そのページから画像を取り除いたページを表示することができる。

コンテンツ変換方式と本システムとの機能の違いを表 3.1 に示す。コンテンツ変換方式では、ユーザの嗜好に合わせて Web ページのコンテンツを簡略化するため、未知のことを調べたりする際に必要なデータが省略されてしまう危険性がある。これに対し、本研究では、PC で閲覧した Web ページから必要な情報を携帯電話に持ち出すことに重点をおき、PC で閲覧している Web ページからユーザが必要とするコンテンツを抜き出す方法を用いている。これにより、元の Web ページがリッチなコンテンツを含むものであろうと、そこからユーザが必要とするコンテンツのみを表示する簡略化された Web ページを作成できる。さらに、その Web ページを携帯電話で閲覧することもできる。

表 3.1: コンテンツ変換方式と本システムの機能の違い

システム名	Web ページのコンテンツ編集	目的のコンテンツが表示される	携帯電話の標準のブラウザでの表示
コンテンツ変換方式			
本システム			

3.2 Web ページの分割提示に関する研究

携帯電話のブラウザで PC 向けに作成された Web ページを閲覧する際の、画面サイズの小ささから生じる問題点に対して、既存のフルブラウザなどでは対策が講じられている。その方法は、予め Web ページの概観を示し、Web ページをいくつかの領域に分け、その領域を選択するとそこが表示される、というものである。そして、その領域の分け方について様々な研究がされている。

Shumeet のシステム [14]は、携帯電話の画面を端末の 1 ~ 9 のボタンに対応させ、9 つの領域に分割する。その際、画面を 9 等分するのではなく、コンテンツのエントロピーを減少させ、決定木を用いて領域を分割する。これにより、文章や画像などの関連する内容が 1 つの領域に収まる分け方ができる。

SmartView [15]や WebThumb [16]は、PDA 向けのブラウザを作成し、そこで分割提示方式を用いている。これらのシステムは画面を決められた数の領域に分割するのではなく、コンテンツの構造を解析し、関連する内容ごとにグループ分けし、そのグループごとに領域分けを行う。そして、ユーザが目的のコンテンツを含む領域を選択すれば、その領域が拡大表示される。

これらの手法を使いこなすには、予め Web ページのレイアウトを把握している必要があり、携帯電話や PDA からのみ Web ブラウジングをするのであれば有用かもしれないが、本研究がターゲットとしている、PC で調べた Web ページの情報を持ち出す、という観点から考えれば、効率のよいものではない。

分割提示方式と本システムとの機能の違いを表 3.2 に示す。この方式は、コンテンツ変換方式と違い、選択された領域のコンテンツは編集せずに全て表示される。さらに、分割提示方式を採用しているのは、携帯電話でもリッチなコンテンツを表示できるフルブラウザや PDA であり、本研究が対象としている一般のブラウザには対応していない。

表 3.2: 分割提示方式と本システムの機能の違い

システム名	Web ページのコンテンツ編集	目的のコンテンツが表示される	携帯電話の標準のブラウザでの表示
分割提示方式	×		×
本システム			

本研究では、ブラウザとフルブラウザを区別し、以降では断りがない限り、ブラウザとは多くの携帯電話に搭載されている標準のブラウザのことを表す。

3.3 Web ページの保存・編集に関するツール

本章の冒頭でも述べたが、日本のインターネット普及率は高く、今後も増加が見込まれる。そこで、インターネットをより快適に利用するために、様々なツールが開発されている。その中で、PCで閲覧している Web ページの情報を効率的に収集・整理・編集できるツールが登場してきている。そのため本節では、そうした中から本研究に関連する Web ページの保存・編集に関するシステムについて述べる。

3.3.1 ScrapBook

ScrapBook[17]は、Web ページの保存と収集したコレクションの整理を手助けするための Firefox の拡張機能である。その機能は主に Web ページの保存と編集に別れ、本研究と関連するのは保存に関する機能である。ScrapBook に搭載されている保存の機能は、Web ページ全体の取り込みと部分的な取り込み、そして、リンク先 Web ページの取り込みがある。

しかし、これらの機能は全て PC 上に保存し、PC 上で編集するためのものであり、携帯電話には対応していない。また、Web ページを部分的に保存する

場合の選択方法は、マウスで保存したいテキストなどを反転させ、それにより選択範囲を指定する。だが、これだと1回の操作で保存できるのは、1つの範囲のみで、Web ページ内の複数箇所を同時に保存することはできない。

3.3.2 紙 copi

紙 copi[18]も ScrapBook 同様、Web ページの保存と編集を行うシステムなのだが、こちらは処理をソフトウェアで行うのでブラウザに依存することはない。また、このシステムも対象としているのは PC のみであり、保存範囲の選択も ScrapBook と同じようにマウスでの範囲指定にて行うので、1つのページの中で複数の範囲を同時に選択することはできない。表 3.3 に本システムとの機能の違いを示す。この中で、「表の切り抜き・再構成」については、ScrapBook にも同じことが言えるのだが、マウスで選択できる範囲であれば、表の中のデータを切り抜くことができる。しかし、同時に複数箇所の切り抜きができないため、表の中の離れた箇所にあるデータを選んで切り抜くことはできない。

表 3.3: 関連ツールと本システムの機能の違い

システム名	ページ内の複数 箇所の切り抜き	表の切り抜き ・再構成	携帯電話での 表示
ScrapBook	×		×
紙 copi	×		×
本システム			

第4章

設計

この章では本システムを実現するための設計目標を示し、目的を達成するために本システムが解決しなければならない問題について議論する。その次に本システムの設計について言及していく。

4.1 システム設計の前提条件

本システムでは、生成したファイルをユーザのローカルディスク上に保存するため、セキュリティの観点から、処理はローカルで行う必要がある。そのため、本研究では、ユーザは Firefox[19]を使用していることを前提として、本システムの処理は Firefox の拡張機能にて行うように設計する。

4.2 Web クリッピングとは

クリッピングとは、特定の図形または領域だけに描画を制限する、という意味であり、そこから派生して新聞・雑誌などから記事を選び取る、という意味でも用いられている。そこで、本システムでは PC で閲覧している Web ページから任意の範囲で切り取るので、この作業を Web クリッピングと呼ぶことにする。なお、以降では本システムのことを WeC (Web Clipping system for mobile phone) と表現する。

4.3 システムの概要

図 4.1 に WeC の概要を示す。WeC は、まず PC で閲覧している Web ページをタグによってブロック表示に変換し、そのブロックを選択することで Web ページから切り抜く範囲を指定する。次に、その選んだブロックから新たな HTML ファイルを生成し、それを投稿・閲覧サイトに投稿する。そして、携帯電話からそのサイトにアクセスすることで、切り抜いた Web ページを閲覧することができる。

4.4 設計方針

本研究で構築するシステムは、Web ページを PC で閲覧しているユーザが、必要な範囲でそれを切り抜き、携帯電話で持ち運びすることを可能にするものである。以下に設計方針を示す。

- 簡単な切り抜き操作

前章の関連研究で示した、既存の Web ページの切り抜きに関するツールでは、切り抜く範囲の指定はマウスでドラッグ&ドロップによる指定方法であった。しかし、それだと 1 つのページ内から複数箇所の切り抜きを同時に行うことができない。そこで WeC では、Web ページから切り抜くことができる範囲をブロックで提示し、そのブロックを選択することで範囲の指定を行う。これにより、ユーザは複数箇所の同時切り抜きが可能になる。

- 元の Web ページのレイアウトを崩さないブロック表示

WeC では、PC で閲覧している Web ページをブロック表示に変換し、そのブロックを選択することで切り抜く範囲を選択する。そのブロック

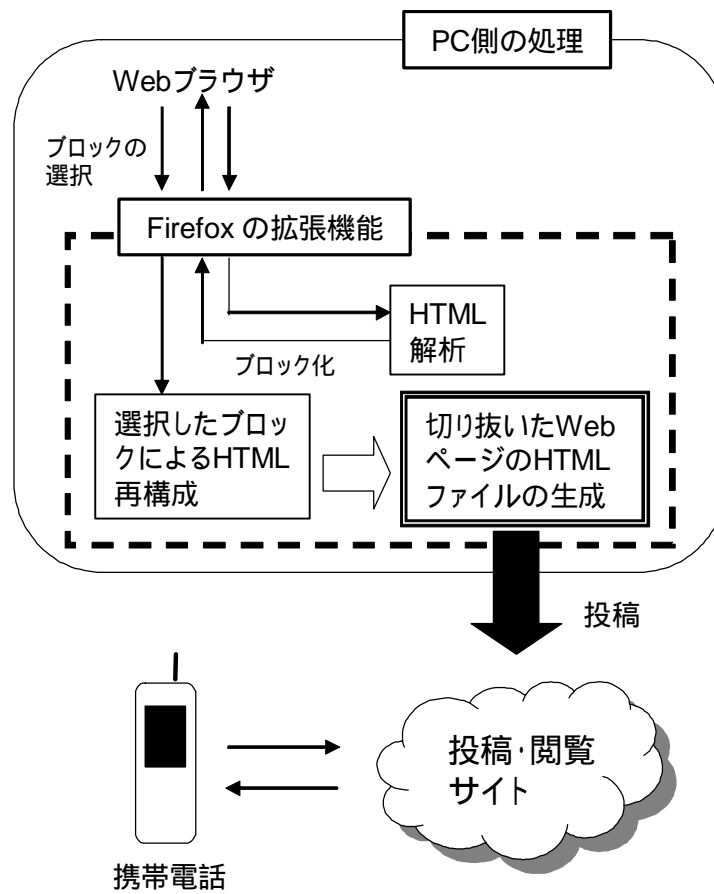


図 4.1: システム概要

表示は、Web ページのタグを判断して実現される。ブロックの描画を追加する際に注意が必要なのは、元の Web ページのレイアウトを崩さないようにする、ということである。本研究の目指すのは、新聞や雑誌と同じように、ユーザが Web ページから必要とするコンテンツを切り抜くことなので、レイアウトを崩すことなくブロックを描画する必要がある。そのため、WeC では HTML をツリー構造で扱い、その構造を崩すことなく、ブロックの描画を行う。

- 見やすい切り抜きの実現

WeC では、切り抜いて作成した新たな Web ページの閲覧を行う端末は携帯電話を想定している。そのため、新たに作成する Web ページのレイアウトは、元の Web ページを考慮せず、クリックしたブロックを順番に縦に並べる方法もある。しかし、ユーザが 1 つの Web ページから複数箇所切り抜いた場合は、切り抜いた順番よりも、元の Web ページのレイアウトの方が記憶に残っている方が多い。よって WeC では、ブロックの選択が終了して、新たな Web ページを生成する時点で、元の Web ページのレイアウトを参考に、レイアウトを再構成する。これにより、切り抜いて完成した Web ページは、PC のブラウザで閲覧したときに、元の Web ページのレイアウトに近いものを閲覧することができる。

- 容易なデータ管理

作成した Web ページを携帯電話に表示させる際に、作成した HTML ファイルを直接携帯電話に送る方法がある。しかし、現在発売されている端末では、HTML ファイルを受信して、それをブラウザで Web ページとして表示することはできない。また、大量に HTML ファイルを作成した

場合、作成のたびに端末にファイルを送信するよりも、どこかにまとめて置いておき、必要なときに必要なファイルだけを端末に送る方がユーザの手間が省ける。そのため、WeCでは投稿・閲覧サイトを設け、そこにファイルをアップロードする手法を用いる。これにより、作成したHTMLファイルを直接携帯電話に送ることができない問題点と、データの管理の煩雑さの両方を解決することができる。

4.5 ブロック化対象タグの決定

前節までで述べたように、本研究が提案する Web クリッピングシステムでは、ブラウザで閲覧している Web ページをタグからブロック表示に変換する。その際のブロック化の対象とするタグの選び方には、次の二通りの方法を考えた。

- あらかじめ、<TABLE>, <DIV>, <H1> - <H6>などのタグを指定しておき、Web ページをこれら指定のタグごとに分割する
- タグは指定せずに、全てのタグをブロック化の対象とする

前者の方法では、切り抜き対象をテーブルなどに限定していれば有効であるが、切り抜き対象を Web ページ内の全コンテンツとした場合には不向きである。一方、後者の方法では、全てのタグをブロック化の対象とすることで、Web ページ内の全てのコンテンツをブロックで囲むことができ、またブロックも動的に変更が可能となるので、今回は後者の方法を用いて実装する。

第 5 章

実装

5.1 実装概要

この章では、本研究で作成した Web クリップングシステムの実装について述べていく。本システムは、切り抜き処理を行う Firefox の拡張機能と、作成した HTML ファイルを保管する投稿・閲覧サイトから構成される。切り抜き処理を拡張機能として実装することで、ユーザは使用しているブラウザ(Firefox)に本研究で作成した拡張機能をプラグインとして追加するだけで、WeC を利用することができる。本研究で作成する拡張機能は Firefox1.5 以降のバージョンに対応しており、約 700 行の JavaScript と XUL のプログラムで構成される。

拡張機能で行う処理は以下の 4 つのパートに分けられる。

- ブロック表示機構
- ブロック選択機構
- ブロックサイズ変更機構
- HTML ファイル作成機構

それらの実装の詳細については、次の節で述べる。

5.2 拡張機能

本研究では、Web ページからの切り抜きに関する処理を Firefox の拡張機能で実装する。そこでの処理は前節で説明した 4 つのパートから構成される。それぞれの詳細について、次の項から述べていく。

5.2.1 ブロック表示機構

まず、ブラウザで閲覧している Web ページをブロック表示に変換する。本機構の設計にあたり、4 章で議論したように、Web ページの HTML から不特定のタグを対象とすることにより、動的にブロックを作成できる方法を採用する。ブロックサイズの自由な変更を可能にするために、Web ページの HTML をツリー構造として考え、その階層ごとにレベルを割り振る。それにより、そのレベルを指定することでブロックサイズを変更することができるようになる。

HTML のツリー構造化

HTML のツリー構造化には DOM (Document Object Model) を用いる。DOM とは、HTML ドキュメントおよび XML ドキュメントのための API (Application Program Interface) である。これは、ドキュメントの論理的構造やドキュメントへのアクセスや操作の方法を定義するものである。これにより、図 5.1 のように Web ページの HTML ソースは DOM ツリーとして、タグなどをノードとするツリー構造に変換し扱うことができるようになる。

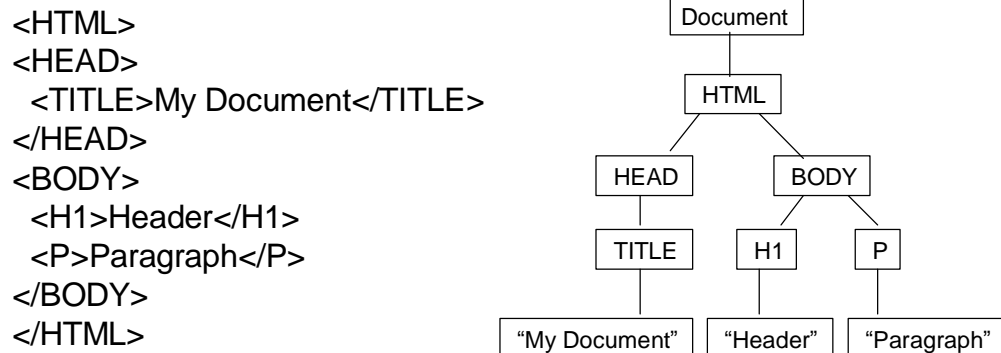


図 5.1: HTML ソースと DOM ツリー

ブロック対象範囲の決定

Web ページの HTML をツリー構造として扱えるようになったところで、次にブロックで囲む範囲の分け方について述べていく。本研究では、ブロックで囲む範囲は DOM ツリーの部分木とする。図 5.2 のように、HTML ソースで `` と `` で挟まれた範囲をブロック化の対象にしたい場合、DOM ツリーで `` をルートとした部分木が同じ範囲を表すことになる。また、部分木をブロック化の対象とすることで、ブロックの動的な変更も容易になる。

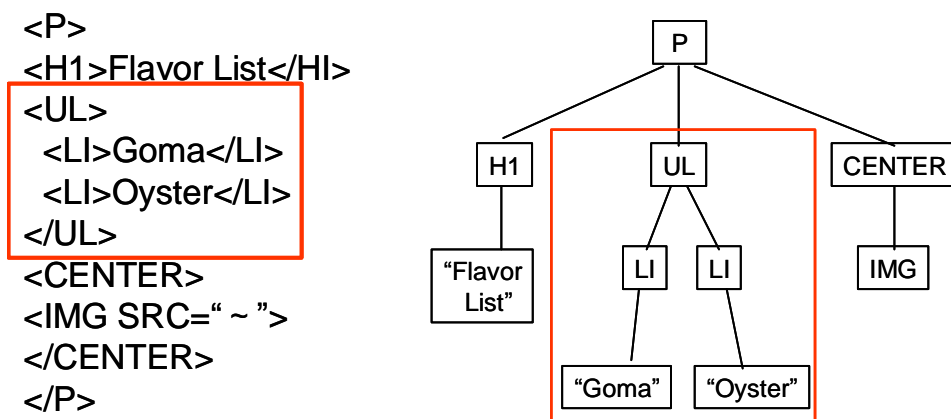


図 5.2: HTML ソースと DOM ツリーの対応

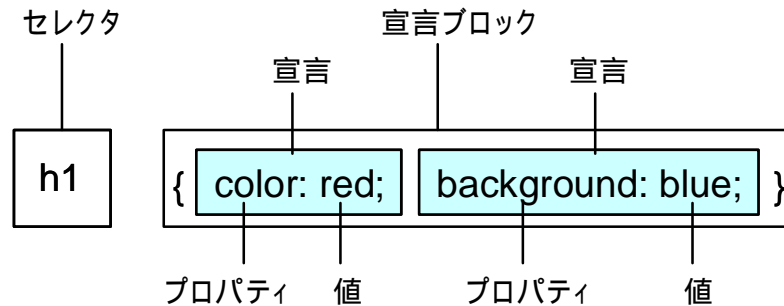


図 5.3: CSS の命令構造

ブロック描画

ブロックの描画は CSS (Cascading Style Sheets) で指定する。CSS とは Web ページのレイアウトを定義する規格である。CSS を使うと、フォントなどの見栄えに関する情報を文章本体から切り離すことができる。レイアウトの定義の方法は、図 5.3 に示す命令構造に従ってスタイルシートに記述する[20]。このセレクトアに見栄えを変更したいタグや id などを設定し、宣言ブロックで具体的に命令を記述する。そのため、ブロックで囲む範囲に該当する部分木のルートに対して、id もしくは class を設定し、それによりブロック描画命令をスタイルシートにて行う必要がある。この id (もしくは class) の設定について、以下の 2 つの方法を考えた。

- 部分木全体を新たに挿入するタグで挟みこみ、このタグに描画命令を設定する
- 部分木のノードのタグに対して直接描画命令を設定する

前者の方法の利点は、新たにタグを挿入するので、その属性として付加する id や class も問題なく設定できることが挙げられる。しかし、この方法では

部分木に含まれるタグに対してスタイルシートで命令が指定されていた場合、ブロック化した際にレイアウトが乱れてしまい、見づらいものになってしまう。そこで、本研究では以下の方法を用いる。

後者では、Web ページの HTML のツリー構造に手を加えないので、ブロック化したとしてもレイアウトが乱れる心配がない。しかし、この方法にも問題点はある。部分木のルートのタグに id や class を設定するのだが、既に設定されてしまっている可能性がある。単純に id などを設定してしまうと、新たに設定したもので上書きされてしまい、Web ページのレイアウトが乱れる原因になってしまう。そこで、設定するものを id ではなく、class に注目してみると、id は 1 つのタグに対して 1 つか設定できないのに対し、class は「class=" , , ”」というようにカンマで区切ることで複数指定できる。そこで WeC では、全てのブロック化対象タグの class 属性に“wec_level, wec_flag, wec_id”という 3 つの属性値を設定する。これらの属性値は以下の意味を持つ。

- wec_level : そのノードの DOM ツリーにおけるレベル
- wec_flag : HTML ファイル作成時のノード保存の判定用
- wec_id : DOM ツリー内のノードの個別の id

詳細については次の項で述べるが、WeC ではブロックの選択の際などに、あらかじめブロックを形成する部分木のルートのタグにイベントハンドラを設定しておく。これにより、ブロックをクリックすることで、選択したブロックに該当する部分木に対してブロックの選択などの処理を行うことができる。ここで、先ほどの id や class と同じように、既に部分木のルートのタグにイベントハンドラが設定されている場合の問題がある。だが、イベントハンドラも 1 つのタグに対して複数指定することができるので問題はない。

5.2.2 ブロック選択機構

ブロックの選択では、まず CSS を使用して、クリックされたブロックの背景と枠線の色の変更をスタイルシートで行う。その際、セレクタとして指定されるのは、前項でタグに設定した class 属性の値の `wec_id` である。この値には、`wec_id5` などのように末尾に数字を付けて設定し、その数字はノードごとに固有のものとする。色の変更を行うことにより、クリックされたブロックが視覚的に分かりやすくなる。スタイルシートで設定する `wec_id` は、選択したブロックに対応する部分木のノードの値とする。これにより、選択したブロックに対して変更を行うことができる。

また、選択と同時に、切り抜きの際の準備もしておく必要がある。次項で、選択したブロックから Web ページを再構成し、新たな HTML ファイルを作成するのだが、再構成の方法として DOM ツリーの再構築を行う。なるべく元の Web ページのレイアウトを保ったまま切り抜くために、選択したブロックに該当する部分木のルートから Web ページ全体のツリーのルートまでのパスは確保しておく。これにより、選択した部分木に関連するタグを確保できるので、元の Web ページのレイアウトをなるべく損なうことなく切り抜きを行うことができる。さらに、複数のタグから構成される表なども表現を損なうことなく切り抜くことができる。パスの確保の方法は、前項で設定した class 属性の値の `wec_flag` を用いる。初期状態では全てのノードに対して `wec_flag0` で設定しておく。ここで指定する値には次の 3 通りがある。

- `wec_flag0`: 初期状態(切り抜き対象外)
- `wec_flag1`: 切り抜きの際に確保されるノード
- `wec_flag2`: 選択された部分木のルート

まず、クリックされたブロックに該当する部分木に対して操作を行う。ここでは、選択した部分木のルートタグの属性値を `wec_flag2` に変更する。これは部分木のルートを表し、DOM ツリーの再構築の際に部分木全体を取り込む目安にする。次に部分木のルートの親ノードに着目する。そのノードの属性値が `wec_flag0` の場合は `wec_flag1` に変更する。この操作を Web ページ全体の DOM ツリーのルートまで繰り返し行う (図 5.4)。

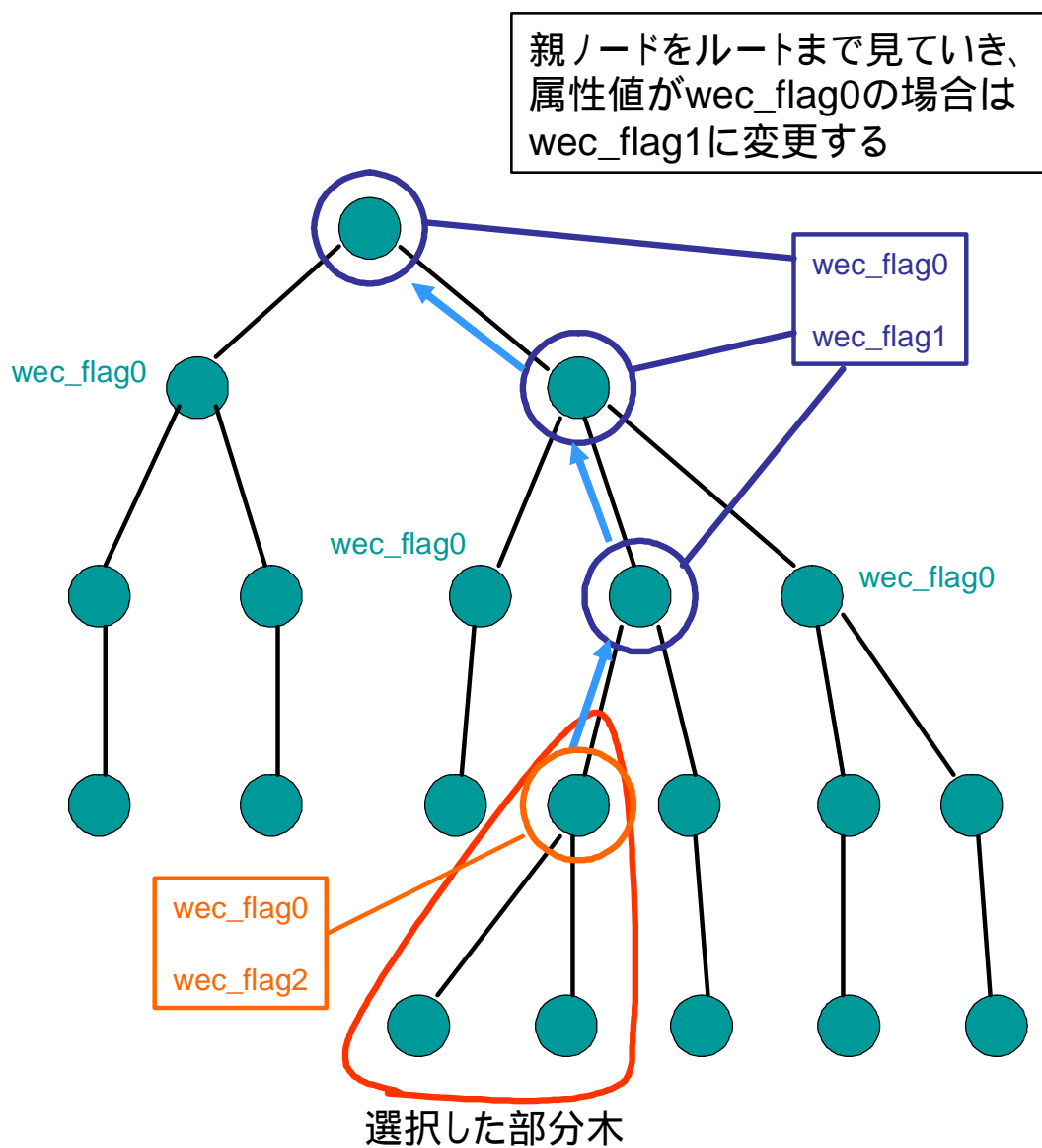


図 5.4: `wec_flag` の値の設定

5.2.3 ブロックサイズ変更機構

ここでは、ブロックサイズの変更を行う。ブロックサイズの変更には、Web ページ内のブロックを一括して行う場合と、局所的にブロックの中に対してだけサイズを小さくする場合の 2 通りを用意した。前項で述べたように、ブロックの描画の命令は部分木のルートのノードに対して行う。これを踏まえて、Web ページ全体の DOM ツリーのルートから各ノードまでの深さによってレベルを設定しておき（図 5.5）、このレベルを変更することで、ブロックサイズを変更できる。このレベルの変更方法によって、一括変更と局所的な変更の両方に対応することができる。例として、wec_level1 のブロック（図 5.6）と wec_level3 のブロック（図 5.7）を示す。図中の赤い枠が本システムの作るブロックである。

ブロックサイズの一括変更

Web ページ内のブロックのサイズを一括で変更した例を図 5.8 に示す。図中の上下の画像を比較すると、Web ページ内の全てのブロックのサイズが変更された様子が見て取れる。この場合のブロックサイズのレベルの変更は、ツリー全体に対して行う。図 5.3 で説明したスタイルシートでの CSS のセレクタはツリーのレベルを表す class 属性の wec_level とする。

ブロックサイズの局所変更

Web ページ内のあるブロックについて、局所的にブロックサイズの変更を行った例を図 5.9 に示す。図中の上下の画像を比較すると、一部のブロックのサイズが変更されたことが分かる。この場合のブロックサイズのレベルの変更は、ツリーの中の部分木に対して行う。スタイルシートでの CSS のセレクタは、局所的に設定を行うので、各ノード固有の属性値の wec_id を用いる。

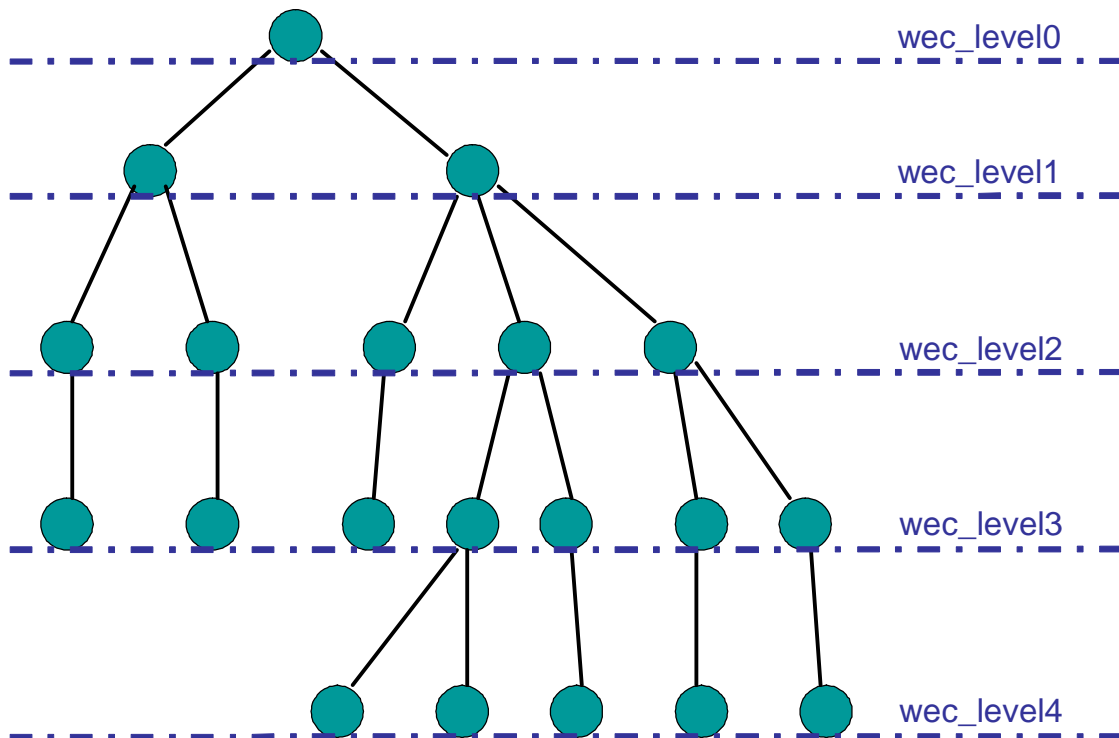


図 5.5: ブロックサイズのレベル設定例



図 5.6: wec_level1 の例



図 5.7: wec_level3 の例

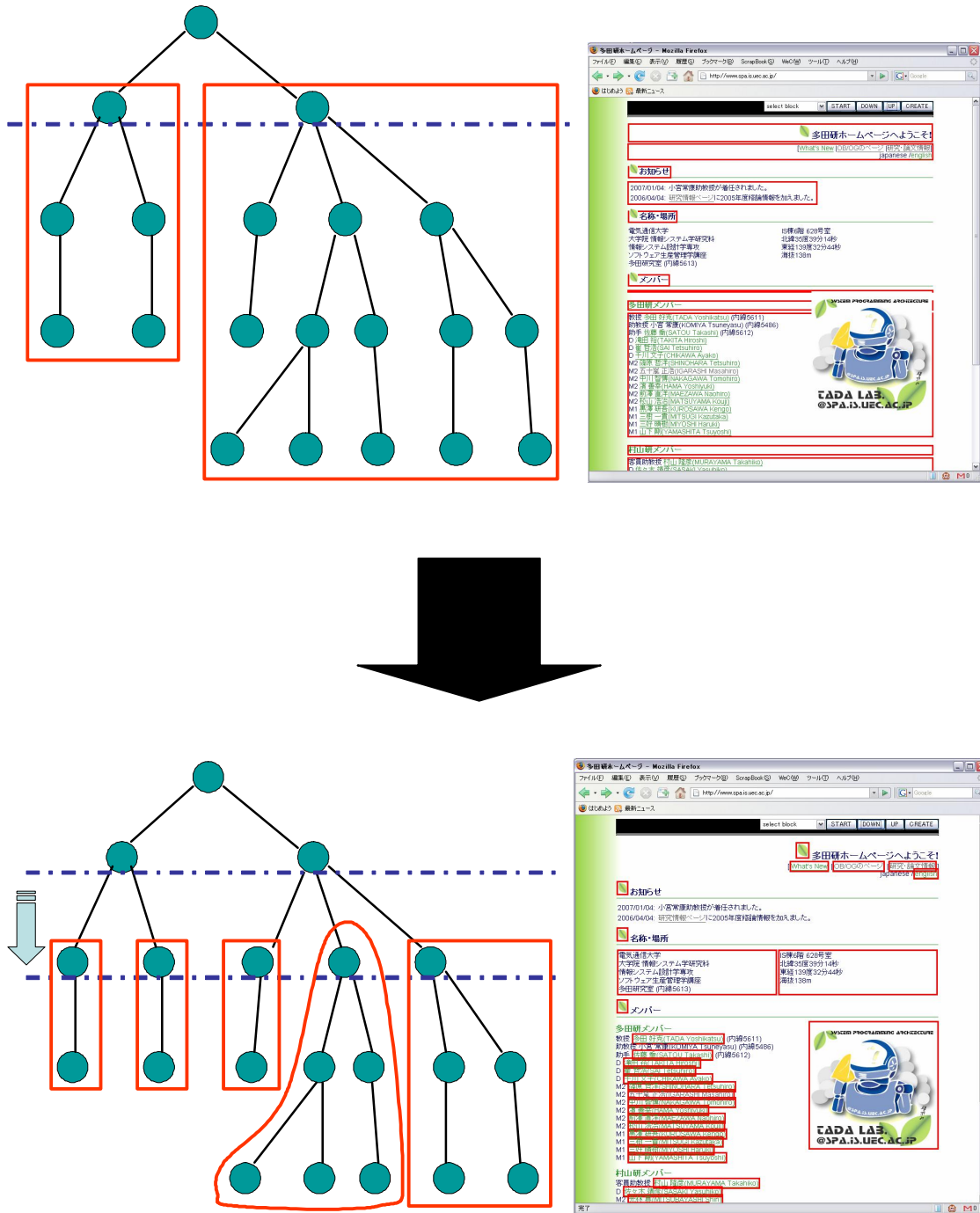


図 5.8: ブロックサイズの一括変更の例

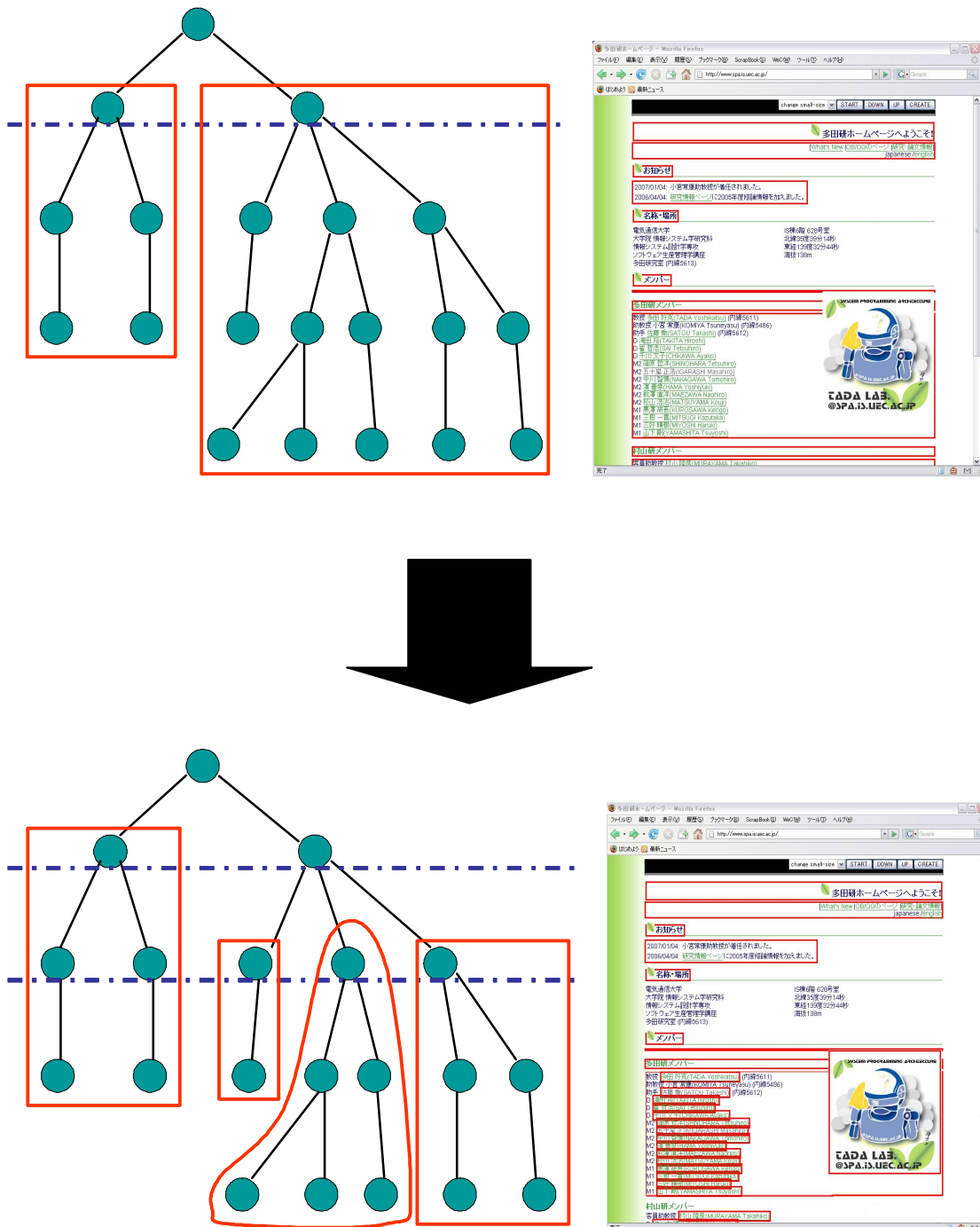


図 5.9: ブロックサイズの局所変更の例

タグを考慮したレベル設定

ブロックサイズのレベルの決定には、ツリー全体のルートから各ノードまでの深さによって決めると述べた。しかし、タグの中にはブロック化の対象から除いてもしまっても支障のないものがある。それには、次のようなものがある。

- <HR>,
のように Web ページの体裁を整えるためのもので、タグ自体に意味を持たないもの
- ツリー構造で表現したときに、親ノード・子ノード共に 1 つしか持たないようなノード (タグ)

前者の場合は、改行や罫線は切り抜く必要がないのでブロック化の対象にしない。後者では、ブロック化したときに、レベルを変更してもブロックサイズは変わらないので、ブロック化の対象からは外す。これにより効果的なブロック表示が実現できる。

5.2.4 切り抜きファイル作成機構

本機構では、前項までによって選択したブロックから切り抜き範囲を決定し、それを元に Web ページから切り抜きを行う。さらに、その際 DOM ツリーを再構築し、WeC のオリジナルツリーを作成する。その後、そのツリーを元に新しい HTML ファイルを生成する。

オリジナルツリーの作成

ここでは、選択したブロックから DOM ツリーの再構築を行い、WeC のオリジナルツリーを作成する(図 5.10)。それにはブロック選択の際に設定した class 属性の値、wec_flag を用いる。設定した際にも述べたが、値には以下のような意味がある。

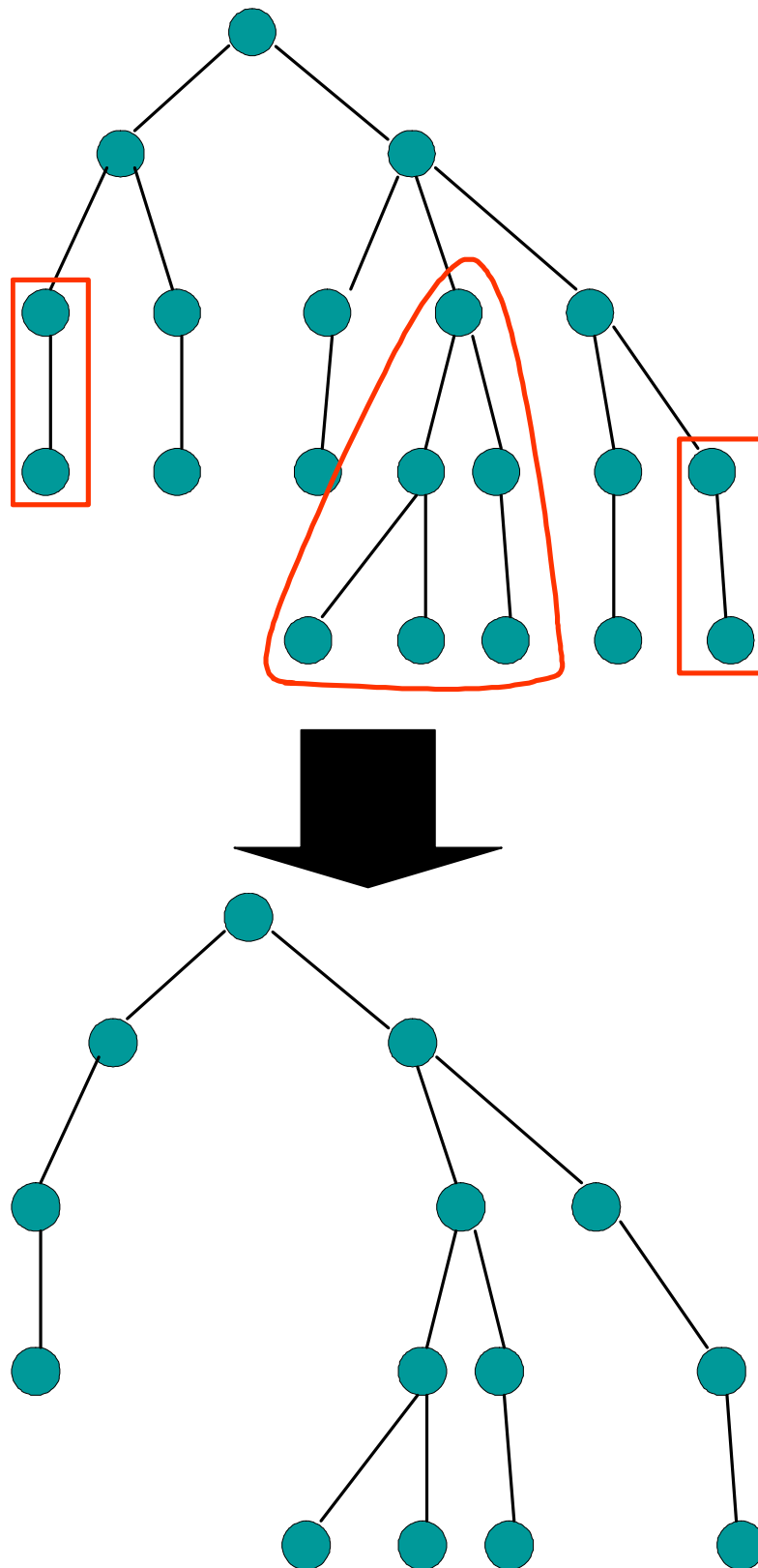


図 5.10: オリジナルツリーの生成

- wec_flag0: 初期状態(切り抜き対象外)
- wec_flag1: 切り抜きの際に確保されるノード
- wec_flag2: 選択された部分木のルート

オリジナルツリーの作成方法としては、まず元の Web ページの DOM ツリーを複製し、オリジナルツリーの原型とする。次に、ルートから幅優先探索を用いて順番にノードの wec_flag の値をチェックしていく。wec_flag0 の場合は、そのノード以下を全て削除する。wec_flag1 であれば、そのノードはオリジナルツリーのルートと部分木を結ぶパスとして必要なので確保する。wec_flag2 の場合は、そのノードをルートとした部分木が選択されたことを表すので、そのノード以下を全て確保する。この操作を全てのノードに対して行うことで、選択したブロックと、その親ノードから構成されるオリジナルツリーが完成する。

HTML ファイルの生成

作成した WeC のオリジナルツリーを HTML ソースの形式に戻し、新たな HTML ファイルとして保存する。これにより、切り抜いた内容から成る新たな Web ページを作成することができる。本システムでは、タグの追加は行わないので、文字の大きさは元の Web ページのサイズが反映される。この HTML ファイルは端末の仕様の都合上、携帯電話に送って Web ページとして表示することができないので、次項で述べる投稿・閲覧サイトに投稿する。

HTML ファイルはユーザのローカルディスク上に保存し、その保存先は図 5.11 のようになる。wec_file とは、WeC が作成するディレクトリで、作成された HTML ファイルはこのディレクトリの中に保存される。wec_file の中にあるディレクトリは、切り抜きを行った年月日を名前としている。そしてこのディレクトリの中に、切り抜きによって生成された HTML ファイルを格納する。ファイル名は、1 つの Web ページから切り抜きによって生成されるファイルは

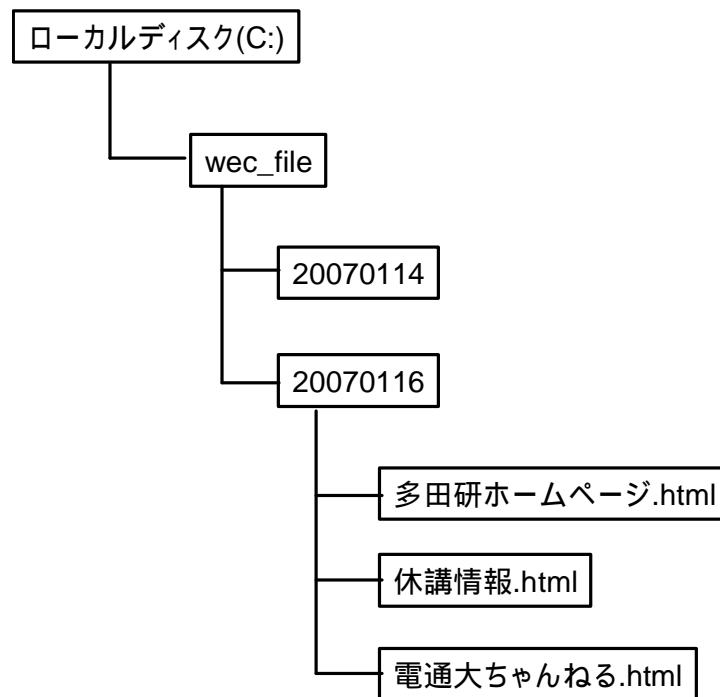


図 5.11: 作成したファイルの保存先例

1つということを考えて、元の Web ページのタイトルにする。これにより、データの管理が容易になる。

5.3 投稿・閲覧サイト

作成した HTML ファイルは、投稿・閲覧サイトに置かれる。本研究で用意した投稿・閲覧サイトは、一般的なアップローダを流用したものである。これは、アップロード機能を有した、Perl で書かれた約 100 行の CGI のプログラムで構成される。ここで投稿されたファイルは一覧表示される。携帯電話でこのサイトにアクセスし、希望のファイルを開くと、切り抜いて作成した Web ページを見ることができる。また、一度端末に表示した Web ページは携帯電話に標準で搭載されている「画面メモ」機能を用いることで、端末に保存することができ、これにより、オフライン時での閲覧も可能になる。

なお、7章ではこのサイトの改善法についても議論する。

第 6 章

評価

6.1 評価方法

この節では、Web クリッピングシステムを用いていくつか Web ページを切り抜く。そして、それを携帯電話のブラウザで表示したときの元の Web ページとの比較を行う。Web ページからの切り抜きは以下のケースを考える。

- ニュースサイトからの記事の切り抜き
- 表を含む Web ページの切り抜き

ニュースサイトは、複雑な HTML 構造をしているため評価の対象とした。また、作成した Web ページは携帯電話で表示させるため、画面サイズの小ささから表示に適していない、表を含む Web ページも評価の対象とした。

それぞれのケースについて、WeC の動作実験を行い、それによって切り抜いてできた Web ページと元の Web ページを携帯電話上で閲覧したときの様子を比較する。

今回の実験では、NTT ドコモが提供している i モード HTML シミュレータ [21]を用いて、携帯電話で表示した際のシミュレートを行う。このシミュレータは、901i シリーズ以降の i モード対応 HTML や Flash、デコメールテンプレート、901iS シリーズ以降の PDF などを表示することができる。

6.2 実験結果

前節で挙げた2パターンのケースそれぞれについて、元のWebページとWeCを用いて切り抜きを行って作成したWebページを、携帯電話のブラウザで表示させたときの違いを見ていく。

6.2.1 ニュースサイトからの切り抜き

この項では、一般的なニュースサイトから写真と記事の切り抜きを行い、元のWebページとの携帯電話上での表示の比較を行う。

まず、切り抜く対象のWebページは、図6.1のようなニュースの記事が書かれたWebページとし、図中の四角で囲った範囲を切り抜く対象とする。このWebページを携帯電話でそのまま表示した画面を図6.2と図6.3に、WeCを用いて切り抜いたWebページを表示した画面を図6.4に示す。

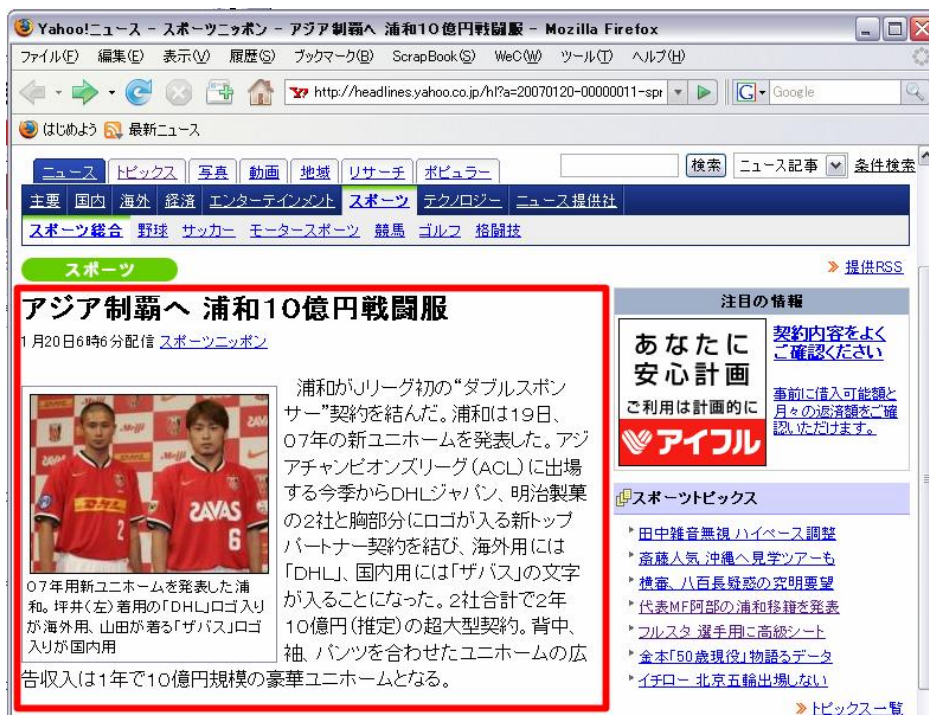


図 6.1: 切り抜く元の Web ページ

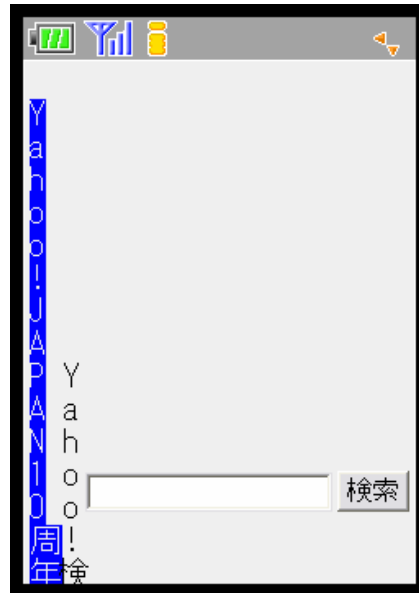


図 6. 2: 図 6.1 の Web ページをそのまま携帯電話で表示した画面

切り抜きを行う前に、この Web ページを携帯電話のブラウザでそのまま表示してみる。図 6.1 の Web ページを携帯電話で閲覧すると、まず表示されるのは、図 6.2 のような画面である。ポータルサイトやニュースサイトは複雑な HTML 構造になっているため、この状態で目的のコンテンツを探すのは困難である。

次に、同じコンテンツを表示したときの違いを図 6.3 と図 6.4 に示す。図 6.3 は、元の Web ページを携帯電話に表示させ、そこから目的のコンテンツを表示させた様子で、図 6.4 は、WeC を使用して目的のコンテンツを切り抜き、新たな Web ページを作成し、それを携帯電話に表示させた画面を表す。



図 6. 3: 図 6.2 で切り抜き範囲に該当するコンテンツを表示した画面



図 6. 4: WeC を用いて作成した Web ページを表示したときの画面

6.2.2 表を含む Web ページの切り抜き

この項では、Web ページ中に表データを含む場合の切り抜きについて議論する。図 6.5 に示す、表を含む Web ページから図中の枠で囲った範囲を切り抜く。

この表をそのまま携帯電話で表示した様子を図 6.6 に示す。携帯電話では、表形式のデータ表現というのは、画面サイズの小ささから、このように端末の画面の幅に合わせて圧縮されてしまい、大変読みづらいものになってしまう。ここで、WeC を用いて、この表データの中から、必要な要素のみを切り抜き、新たな表を生成する。例として、5 段ある列の中から 2 つを選択し、その中からさらに要素を選択し、表を再構成し、携帯電話に表示した画面を図 6.7 に示す。

開催日	名称	論文等応募締切日	参加締切日	開催地
	平成19年度会議モータ基礎	2月28日(火)		
	論文誌「次世代社会基盤をもたらす高度交通システムとモバイル通信システム」特集への論文投稿	3月31日(土)		
	論文誌「インタラクションの理解とデザイン」特集への論文投稿	3月31日(土)		
	論文誌「人間中心のユニバーサル/コネクタネットワークサービス」特集への論文投稿	4月6日(金)		
	論文誌「情報洪水時代のネットワークサービス」特集への論文投稿	5月11日(金)		
	論文誌「社会的課題に挑む情報システム」特集への論文投稿	5月15日(火)		
	論文誌「新しいパラダイムの中での分散システム/インターネット運用・管理」特集への論文投稿	5月28日(月)		
	論文誌「インベションプロセスのスタディ」特集への論文投稿	6月30日(土)		
1月23日(火)~1月26日(金)	Asia and South Pacific Design Automation Conference 2007 (ASP-DAC2007)	7月10日(月)		パシフィコ横浜
1月25日(木)	ソフトウェアジャパン2007		1月18日(木) 当日可	タワーホール船堀
1月25日(木)	ソフトウェアジャパン2007		1月18日(木) 当日可	タワーホール船堀
1月25日(木)~1月26日(金)	ウィンターワークショップ 2007・イン・那覇			ホテルオーシャン、他
1月25日(木)~1月26日(金)	第141回データベースシステム・第62回グループウェアとネットワークサービス・第16回放送コンピュティング研究グループ合同研究発表会	11月24日(金)	当日のみ	メーブル有馬(神戸市有馬温泉)
1月26日(金)	北陸支部主催研究講演会「次世代ネットワーク		参加費無料・学会員	富山県立大学(富山県津幡町)

図 6.5: 表を含む Web ページの例

開催日	名称	論文等応募締切日	参加締切日	開催地
	平成19年度会誌モニタ夕募集	2月28日(水)		

図 6.6: 図 6.5 の Web ページを携帯電話で閲覧した画面

開催日	名称
1月23日(火) ~ 1月26日(金)	Asia and South Pacific Design Automation Conference 2007 (ASP-DAC2007)
1月25日(木)	ソフトウェアアジアジャパン2007
1月25日(木)	ソフトウェアアジアジャパン2007

図 6.7: WeC を用いて表を編集した切り抜き例

6.3 考察

6.3.1 本システムで作成した Web ページの有効性

前節で、いくつかのケースで WeC を用いて切り抜きを行い、元の Web ページとの携帯電話上での表示の比較を行った。

まず、ニュースサイトから記事と写真を切り抜き、元の Web ページと比較する。それは、図 6.3 と図 6.4 を比較することで分かる。図 6.4 が WeC を用いて、目的のコンテンツのみを切り抜いて作成した Web ページである。一方、図 6.3 は元の Web ページをそのまま表示したもので、ページ全体のレイアウトに影響されてしまい、可読性が低下してしまっている。さらに、図 6.2 は携帯電話で元の Web ページを表示した際の最初の状態である。同じ Web ページを PC のブラウザで表示した様子は図 6.1 のようになる。このような様々なコンテンツが 1 つのページ内に存在する、複雑な HTML 構造をした Web ページは、携帯電話のブラウザで表示すると図 6.2 のようになってしまい、目的のコンテンツを探し出すのは困難である。WeC を用いて作成した Web ページは、目的のコンテンツから構成されるので、図 6.4 のように大変見やすいものになる。また、図 6.4 は、この Web ページを表示した最初の状態である。このことから、Web ページ表示の初期段階からも WeC が有効であると言える。

次に、表データを含む Web ページの切り抜きについて議論する。図 6.5 のような Web ページ中の表から一部を抜き出し、表を再構成して作成した Web ページと元の Web ページの表示の比較は、図 6.6 と図 6.7 にて行える。表形式のデータ表現を携帯電話で行うと、図 6.6 のように、表の幅が画面の幅に合わせて圧縮されてしまい、表内部のデータの折り返しが増えて可読性が低下してしまう[22]。この表のデータの中から、ユーザが必要とする情報が一部だったときに、WeC を用いて必要なデータから成る表を新たに作成することができる。今回の例では、元の Web ページでは 5 段ある列から、2 段だけ必要とし、

そこからデータを選び、表を作成し、表示した結果が図 6.7 である。切り抜く前よりも可読性が増したの明らかである。

以上のことから、PC で見た Web ページを切り抜いて携帯電話で持ち出す、という本研究の目的は達成され、さらにこの方法が画面サイズの小さい携帯電話で有効であることが分かった。

6.3.2 携帯電話に対応しているマークアップ言語

評価実験で表を含む Web ページの切り取りを行った際に、シミュレータの他に実機での検証も行ったところ、表が表示されない端末があった。これは、切り抜く元の Web ページを表示したときも同じで、キャリアや端末によって対応しているマークアップ言語が異なることが原因である。文献[23][24][25]を参考に携帯電話の各キャリアの端末の対応しているマークアップ言語について見ていく。対応言語とその特徴を表 6.1 に示す。表に示した言語の他に、DoCoMo も SoftBank も XHTML に対応している。XHTML Basic とは、W3C で勧告された、今後の携帯電話端末向けマークアップ言語のベースとなるものである。

表 6.1: 各キャリアの端末の対応言語とその特徴

キャリア名	対応言語	特徴
DoCoMo	iモード対応 HTML	バージョンごとに対応するタグが拡張 テーブルは Ver.6.0 から対応
au	XHTML Basic	XHTML のサブセットで携帯端末向けに作成
SoftBank	ソフトバンク携 帯向け HTML	HTML のサブセット iモード対応 HTML と同等

XHTML は HTML を XML の規則の中で再定義したものであり、HTML のみにしか対応しないブラウザでもある程度表示することが可能となる。ユーザの使用する端末がこれらの言語に対応しない古いタイプであった場合には、WeC を用いても表などを表示できない可能性もある。しかし、新しい機種はこれらの言語に対応しており、今後 WeC の使用できる環境は増えていくと考えられる。

第7章

問題点と課題

本研究では、Web ページからユーザが任意の範囲で切り抜きができる Web クリップングシステムを提案した。WeC では切り抜きの際に、Web ページの HTML をツリー構造に置き換えて考えているのだが、これによる問題点とシステムとしての今後の課題について以下に述べる。

7.1 切り抜き精度に関する問題

WeC は、DOM を用いて Web ページの HTML をツリー構造として扱い、ツリーの部分木をブロックで選択することで切り抜き範囲の選択を行った。だが、この方法では扱うことができる最小単位はツリーのノードになってしまい、それ以上の細かい範囲の選択はできない。図 7.1 を例にとって見てみると、<P>タグで指定された段落の中にテキストとリンクが記されている。図の右側のツリー構造にある 2 つの A はアンカータグのことであり、画面中の 2 箇所のリンクを表す。その上下にある #text はテキストノードであり、本文がリンクの前後で分けられ、それぞれに格納されている。WeC で扱う最小単位は 1 つのノードなので、この画面ではアンカータグに挟まれたテキスト（「リサーチ図形」など）は切り抜くことができる。しかし、それ以外の文章では、1 つのノードの中にまとめて格納されてしまっているため、文章の一部を切り抜くことは

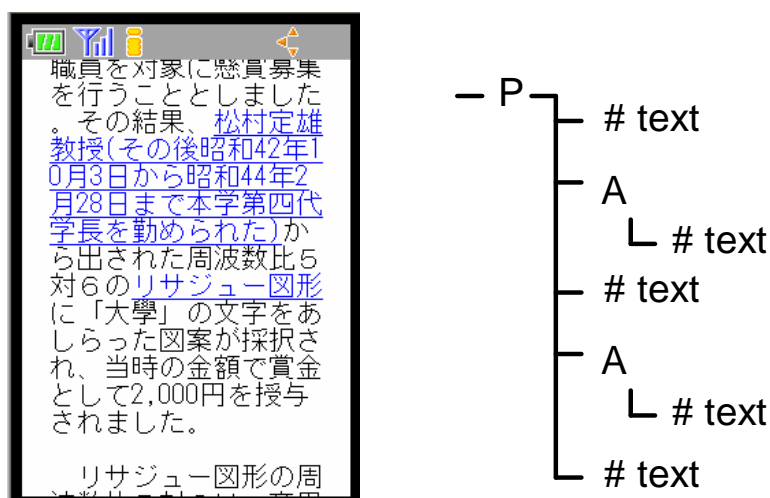


図 7.1: Web ページ中のテキストとそれに対応するツリー構造

できない。そのため、このノードのテキストの中の一部を切り抜くことは現段階の WeC ではできない。4 章で紹介した ScrapBook などのツールは、マウスで切り抜く範囲を指定する方法を用いているのだが、この問題解決のためには、この方法を部分的に取り入れることが有効であると思われる。

7.2 初期表示のブロックサイズについて

WeC では、全体のツリーのルートから各ノードまでの深さによってブロックサイズのレベルを設定した。システムを起動すると、ブロックが表示されるのだが、初期表示は予め設定した一定のレベルである。しかし、ツリーの構造は完全二分木のようになることはなく、偏りがあることが多い。また、<TABLE>などが含まれる場合、初期表示はテーブルの要素ではなくテーブル全体をブロック化した方が扱いやすく、HTML タグの種類によっても初期表示の優先度を考慮する必要があると考えられる。これらのことから、ツリーの構造やタグの種類によって初期表示のレベルを個別に設定する方法を考える必要がある。その際、ブロックサイズの変更方法は図 7.2 のように、レベルで行うのではなく、初期表示の段階のレベルを元に行うのがよいと考えられる。

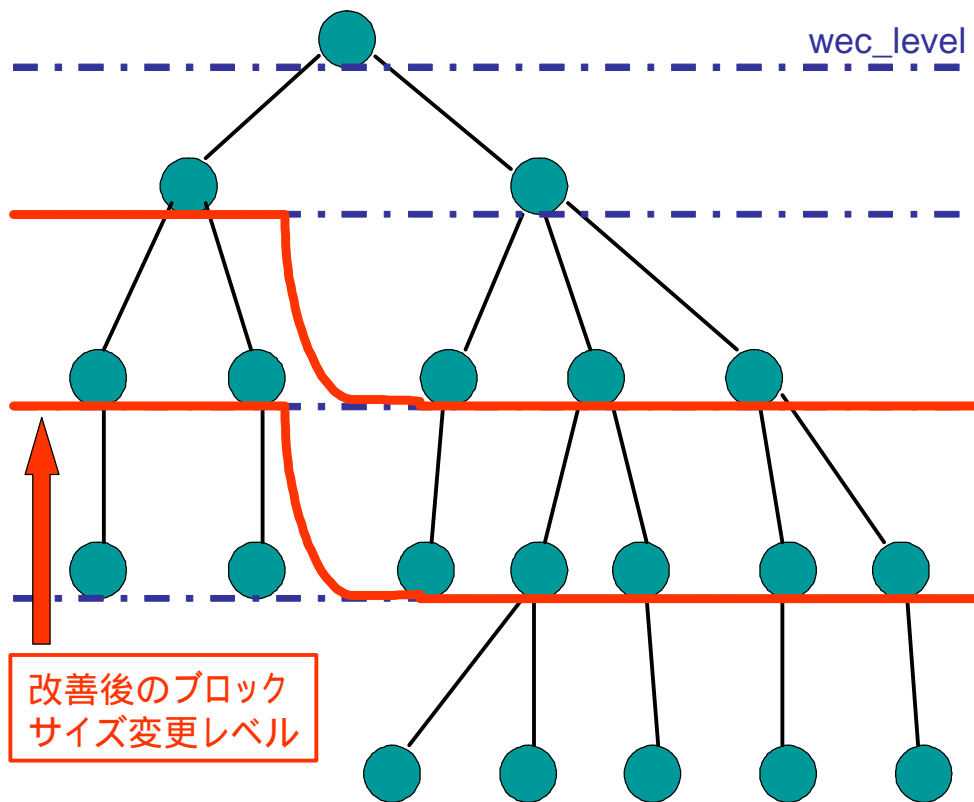


図 7.2: 初期表示のレベルによるブロックサイズの変更例

7.3 フレームを用いた Web ページへの対応

現在、WeC はフレームを用いた Web ページには対応していない。フレームを使ってない Web ページは、<HTML>タグから始まり、その中で HEAD と BODY が展開される構造になっている。WeC は DOM を用いて、この HEAD 部分と BODY 部分に変更を加えている。しかし、フレームを用いた Web ページは、例えば 2 つのフレームから構成される場合、図 7.3 に示す DOM ツリーのようなになる。この場合、<FRAME>タグ以下で別の Web ページが展開されるので、フレームを使わない Web ページと同じ方法で処理を試みてもエラーが起きてしまう。この問題を解決するためには、処理を行う DOM ツリーのノードの指定方法に工夫が必要である。

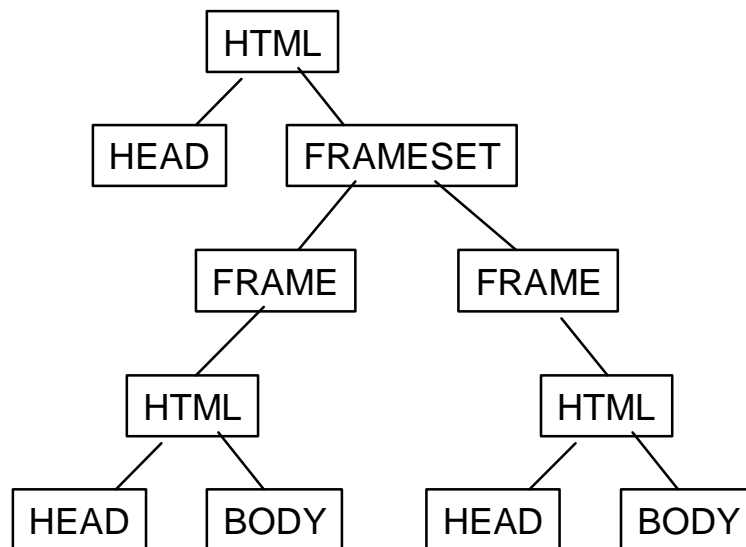


図 7.3: フレームを用いた Web ページの DOM ツリーの例

7.4 機能の拡張性

WeC は、作成した HTML ファイルをユーザのローカルディスク上に保存する。このことから、セキュリティの面などから考慮して、処理はユーザサイドで行うことができるように、Firefox の拡張機能にて実装を行った。しかし、この方法ではユーザが PC で使用できるブラウザを制限してしまうことになってしまう。そこで、切り抜き等の処理をブラウザではなく、プロキシサーバを立てて、そちらで行う方法が考えられる。サーバをローカルで立てるようになることで、セキュリティの心配もいらない。これにより、ユーザが使用するブラウザに依存することなく、WeC の機能を利用することができるようになる。

7.5 投稿・閲覧サイトの改善

5 章でも述べたように、本研究で用意した投稿・閲覧サイトは単純なアップローダの機能しか持たない。しかし、投稿したデータの管理については、まだまだ改善の余地があり、その改善により、更なる利便性の向上が見込まれる。考えられる改善としては、このサイトを PC と携帯電話からアクセスしたときに振り分けて、それぞれに適したレイアウトのサイトを表示させることや、データ一覧の提示方法、カテゴリ分けなどが挙げられる。

7.6 切り抜いた Web ページの融合

今後の展望としては、切り取った Web ページの融合が考えられる。例えば、2 章の利用シーンで寺の情報をいくつか調べたときに、ある寺について複数の Web ページから情報を切り抜いたとする。その場合には、Web ページの分だけ HTML ファイルを作成するよりも、「寺の情報」として、1 つの HTML ファイルに情報をまとめてしまった方が有効であると考えられる。そのためには、選択したコンテンツから 1 つのツリーを作成する必要がある。

第 8 章

おわりに

本研究では、Web ページをユーザの任意の範囲で切り抜くことができる Web クリップングシステムを設計し、実装した。携帯電話での Web ブラウジングが画面サイズの小ささから困難なものになるという問題点と、PC で調べた Web ページの情報を持ち出す際に、紙に印刷した場合に生じる問題点を解決するために、Web ページから必要なコンテンツを切り抜いて携帯電話に表示させる方法を提案した。これにより、ユーザはインターネットで調べた情報から必要なものだけを切り抜き、携帯電話で持ち出すことができる。

効果的な切り抜きを実現するために、本研究で作成したシステムでは、ブラウザで閲覧している Web ページの HTML をツリー構造で考え、その部分木を元に Web ページをブロック表示に変換する。そして、そのブロックを選択することで切り抜き範囲を決定することができる。また、ブロックはサイズの変更が可能で、Web ページ内のブロックを一括して変更する方法と、個別のブロックに対して内部を細かくする方法を用意した。

WeC を用いて、数パターンの Web ページに対して切り抜きを行い、携帯電話で表示したときの、元の Web ページとの比較を行った。その結果、ニュースサイトから記事と写真を切り抜いた際には、目的のコンテンツのみから構成される、WeC が作成した Web ページの方が携帯電話での閲覧には適していることが分かった。また、表を含む Web ページについては、元の Web ページを

携帯電話に表示させると、画面サイズの問題から可読性が低下してしまう。ここで、WeCを用いて、表データの中から必要なデータを切り抜き、再構成した表を携帯電話で表示すると、可読性が増すことも結果として得られた。

謝辞

本研究を遂行するにあたっては、たくさんの方々にお世話になりました。

まず、指導教員の多田好克先生には日頃から熱心なご指導、そしてご鞭撻を賜りました。さらに本研究のシステムについて、発想の段階からお付き合いいただきました。また、ご多忙中にもかかわらず論文の草稿を丁寧に読んで下さり、大変貴重なご助言をいただきました。また、研究を進めるにあたり、佐藤喬助手には貴重な助言をいただきました。ここに厚く御礼申し上げます。

そして、本研究が行えたことは、研究方針や方法論について議論をし、共に研究生活を送ってきた多田研、村山研の学生諸氏のおかげでもあります。最後にこれらの皆さんに感謝いたします。

参考文献

- [1] 社団法人 電気通信事業者協会: “携帯電話/IP 接続サービス(携帯)/PHS/無線呼出し契約数,” 2006.
- [2] 総務省 情報通信政策局: “平成 17 年 通信利用動向調査報告書,” 2006.
- [3] Opera.
<http://jp.opera.com/products/mobile/>.
- [4] jig browser.
<http://br.jig.jp/pc/>.
- [5] NTT ドコモ: “NTT ドコモレポート No.40 携帯電話のデータ通信速度の進化,” 2006.
- [6] 財団法人インターネット協会: “インターネット白書 2006,” 2006.
- [7] 内閣府 経済社会総合研究所景気統計部: “消費動向調査,” 2006.
- [8] C.R.Anderson, P.Domingos, and D.S.Weld: “Personalizing web sites for mobile users,” Proc. World Wide Web Conference(WWW'01), pp.565 -575, 2001.
- [9] M. Jones, G. Marsden, N. Mohd -Nasir, and G. Buchanan: “A site -based outliner for small screen web access,” World Wide Web Conference (WWW'99), 1999.
- [10]M. Jones, G. Buchanan, and N. Mohd -Nasir: “An evaluation of WebTwig - a site outliner for handheld Web access,” International Symposium on Handheld and Ubiquitous Computing (HUC'99), pp. 343-345, 1999.
- [11]M. Jones, G. Marsden, N. Mohd -Nasir, K. Boone, and G. Buchanan: “Improving Web Interaction on Small Displays,” World Wide Web Conference (WWW'99), pp. 1129-1137, 1999.

-
- [12] O. Buyukkokten, H. Garcia-Molina, A. Paepcke, and T. Winograd: "Power browser; Efficient web browsing for PDAs," Proc. Human-Computer Interaction Conference 2000 (CHI 2000), vol. 2, pp. 430-437, 2000.
- [13] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke: "Seeing the whole in parts: Text summarization for web browsing on handheld devices," Proc. 10th World Wide Web Conference (WWW'01), pp. 652-662, 2001.
- [14] Shumeet Baluja: "Browsing on Small Screen: Recasting Web-Page Segmentation into an Efficient Machine Learning Framework," World Wide Web Conference (WWW2006), 2006.
- [15] Natasa Milic-Frayling and Ralph Sommerer: "SmartView: Enhanced Document Viewer for Mobile Devices," Microsoft Research, MSR-TR-2002-114, 2002.
- [16] Jacob O. Wobbrock, Jodi Forlizzi, Scott E. Hudson and Brad A. Myers: "WebThumb: interaction Techniques for Small-Screen Browsers", Proceedings of the 15th annual ACM symposium on User interface software and technology, pp. 205-208, 2002.
- [17] ScrapBook.
<http://amb.vis.ne.jp/mozilla/scrapbook/>.
- [18] 紙 copi.
<http://www.kamilabo.jp/>.
- [19] Mozilla Japan.
<http://www.mozilla-japan.org/>.
- [20] Eric A. Meyer: "CSS 完全ガイド," オライリー・ジャパン, 2005.
- [21] NTT ドコモ: "作ろう i モードコンテンツ: i モード HTML シミュレーター,"
<http://www.nttdocomo.co.jp/service/imode/make/content/html/tool2.htm>
- [22] 安富 大輔, 増田 英孝, 中川 裕志: "携帯端末画面への HTML 表データの表示方法," 情報処理学会デジタルドキュメント研究会, No. 81, pp. 17-24, 2001.

[23] NTT ドコモ: “ 作ろう i モードコンテンツ, ”

<http://www.nttdocomo.co.jp/service/imode/make/>.

[24] KDDI au: “ EZfactory, ”

<http://www.au.kddi.com/ezfactory/index.html>.

[25] ソフトバンクモバイル株式会社: “ ウェブコンテンツ開発ガイド, ” 2006.

付録 A 本システムの操作方法

ここでは、実際に本システムを使用するにあたって必要な操作方法の説明を行う。例として、筆者が所属する研究室の HP において切り抜きを行っていく。

A.1 本システムの画面構成

本システムを起動した状態では、ウィンドウはメニューエリアと本文表示エリアの 2 つから構成される（図 A.1）。

メニューエリア

メニューエリアには、本システムの操作に関するメニューが表示される（図 A.2）。切り抜き操作では、このメニューで動作を選択することにより、マウスで行える操作の変更ができる。表示されるメニューは以下の項目で構成される。

- プルダウンメニュー

ブロックをクリックしたときの動作をここで選択できる。選択項目は「select block」、「change small size」、「change large size」の 3 通り。それぞれの機能は、切り抜きを行うブロックの選択と、クリックしたブロックを対象としたブロックサイズの変更を行える。

- 「UP」ボタン

Web ページ全体のブロックサイズを変更できる。このボタンでは、レベルを 1 段階上げることができる。

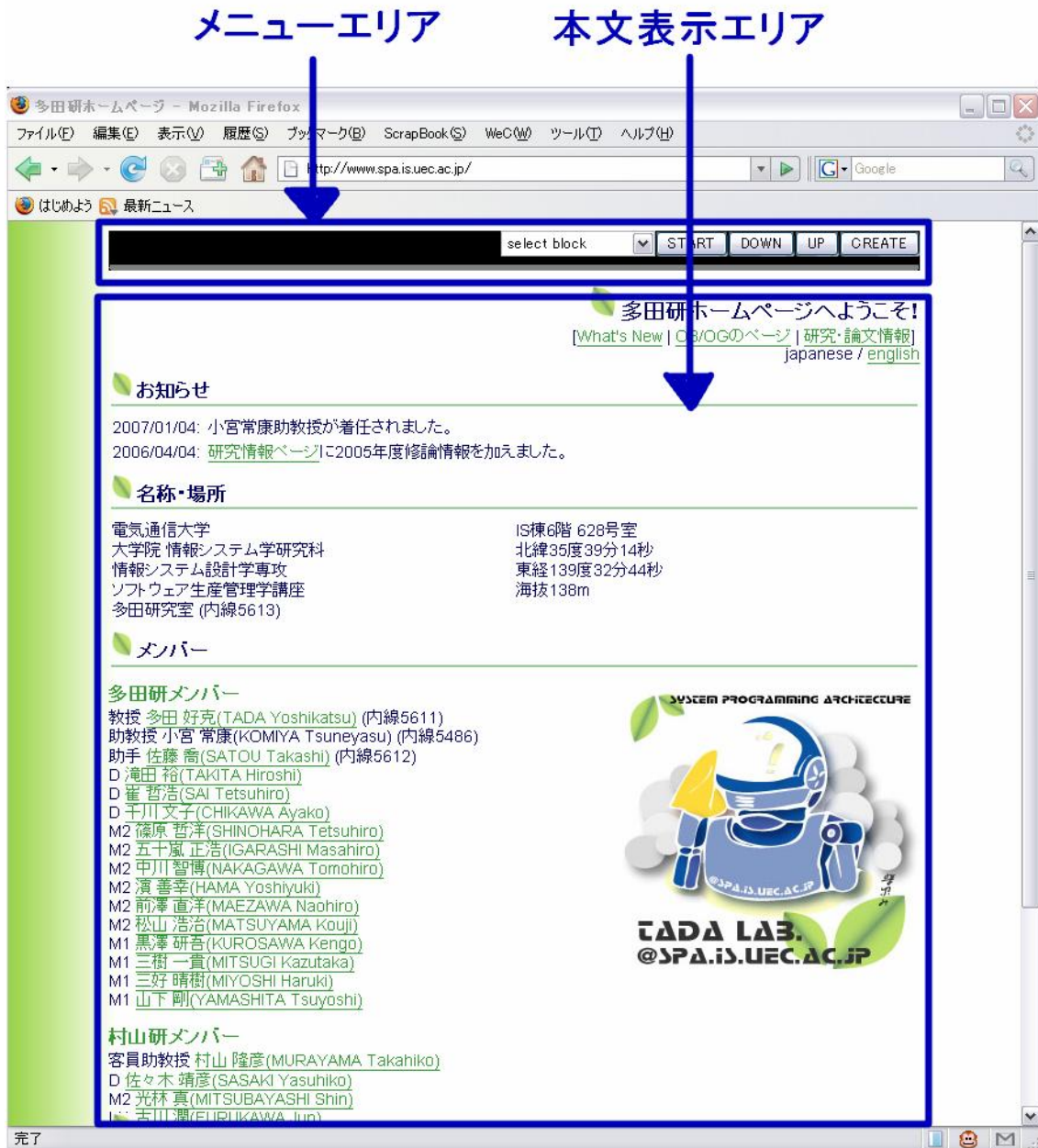


図 A.1: 本システムの画面構成



図 A.2: メニューバー

- 「DOWN」ボタン
Web ページ全体のブロックサイズを変更できる。このボタンでは、レベルを 1 段階下げることができる。
- 「CREATE」ボタン
このボタンを押すと、選択したブロックから新たな HTML ファイルを生成する。

本文表示エリア

元々閲覧していた Web ページは、この本文表示エリアで表示され、メニューエリアでの操作により、ブロック表示に変換される。

A.2 起動方法

まず、切り抜きたい Web ページを PC に表示する。次に、ウィンドウ上部のメニューバーが右クリックで表示されるメニューの中から、「WeC 起動」をクリックする。すると、Web ページ上部にメニューエリアが表示され、ウィンドウが図 A.1 のような画面構成になる。その後、START ボタンを押すと、初期設定のレベルのブロック表示に変換される。

A.3 選択方法

切り抜きたい範囲の指定方法は、この表示されているブロックを選択することで行う。選択方法は表示されているブロックをクリックするだけでよく、

1つの Web ページ内であれば、ブロックはいくつでも選択することが可能である。

A.4 ブロックサイズの変更方法

選択するブロックのサイズは変更することができる。その変更には以下の 2 通りの方法がある。

A.4.1 全体的な変更

Web ページ内のブロック全てを一括して変更する場合は、メニューエリアに表示された「DOWN」、「UP」ボタンによって、ブロック表示のレベルを変更することができる。閲覧している状態から「DOWN」ボタンを押した例を図 A.3 に示す。

A.4.2 部分的な変更

Web ページ内のあるブロックについて、そのブロックの中だけをさらに細かいサイズのブロックに変更することもできる。その場合は、メニューエリアの左端のプルダウンメニューから、「change small size」、「change large size」を選択し、対象のブロックをクリックすると、クリックしたブロックに対してブロックサイズの変更を行うことができる。実際に、1つのブロックの中だけを細かくした例を図 A.4 に示す。

A.5 HTML ファイル生成方法

切り抜きたいブロックの選択が終了した時点で、メニューエリア右端の「CREATE」ボタンを押すと、切り抜きから構成される新たな HTML ファイルの生成が開始される。生成された HTML ファイルは、ユーザのローカル

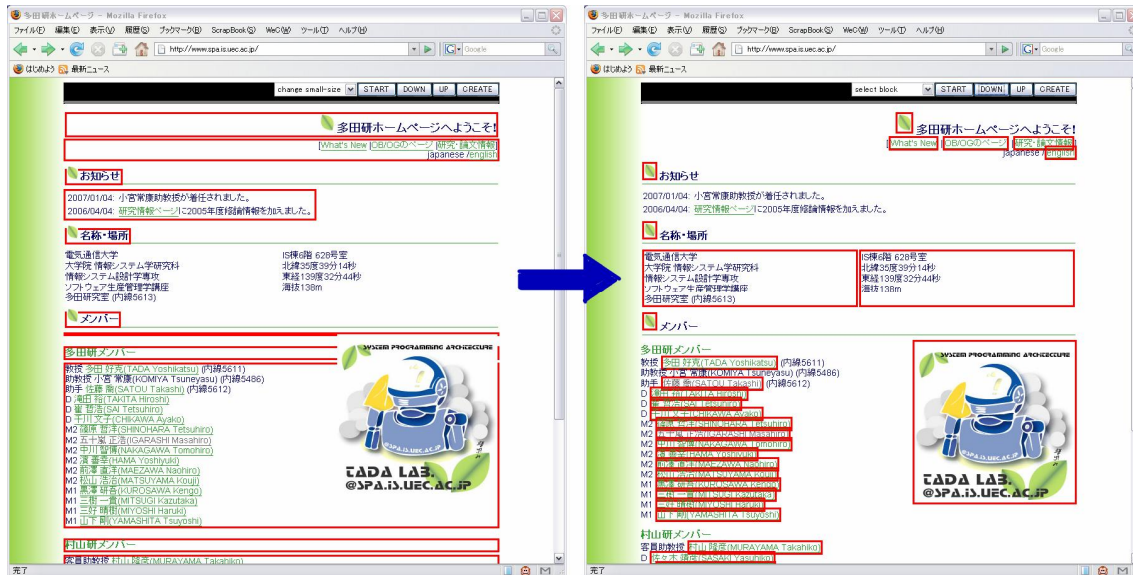


図 A.3: ページ全体のブロックサイズの変更例

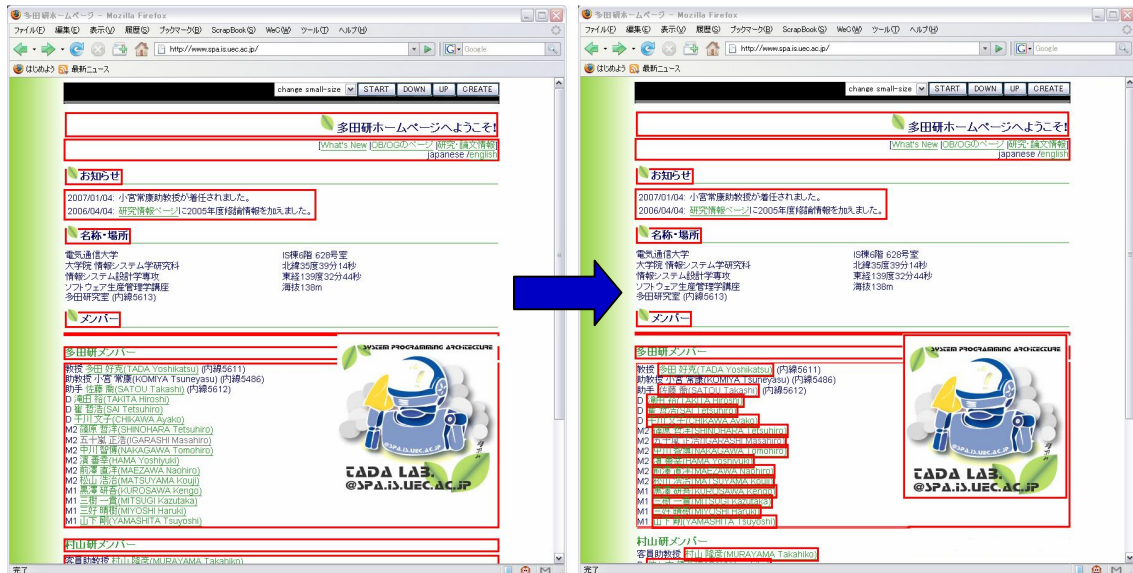


図 A.4: ページの部分的なブロックサイズの変更例

ディスク上に保存される。保存場所は、“ C:¥wec_file¥ (ファイル作成日時を名前としたディレクトリ) ¥作成した HTML ファイル ” となる。また、ファイル名は「切り抜いた元の Web ページのタイトル .html」とする。

A.6 携帯電話での閲覧方法

まず、メニューバーの「WeC」をクリックすると、メニュー一覧が表示され、その最下段に「投稿サイト」というがあるので、それをクリックする。すると、新しいタブに投稿・閲覧サイトが表示されるので、こちらで投稿する。

投稿後、携帯電話から同サイトにアクセスし、一覧表示された HTML ファイルの中から、そのとき必要なファイルを開くことで閲覧できる。また、一度表示した Web ページは携帯電話の「画面メモ」などの機能を用いることで、端末に保存することができ、オフライン時での閲覧も可能となる。

付録 B 著作権について

本システムを使用すれば、インターネット上の情報を簡単に自分の PC の中に取り込むことができ、またそれを携帯電話に持ち出すこともできる。しかし、インターネット上で公開されている情報の多くは、公開している Web ページを作成した作者が著作権を有しており、この権利を侵害することのないような取り扱いが必要になる。著作権法では、以下のような利用が認められているので、参考にしていきたい。

「私的使用のための複製」

個人的な用途等の私的な使用範囲を想定した著作物の複製は、著作者の許諾がなくてもこれを行うことができる。

「引用」

第三者に発表するレポートや広く公衆に閲覧させるホームページ等に他人の著作物を引用して利用したい場合、著作権法第 32 条で定める「正当な引用」の範囲内であれば、著作者の許諾を得ることなく、利用することができる。

一般に他人の著作物の引用については、以下のような要件をすべて満たす必要があると言われている。

- その引用が必要不可欠であること
- 引用する他人の著作物があくまでも「従」であって、「主」ではないこと
- 引用する著作物と自己の著作物とが明確に区別されていること

- 引用する量は必要最小限であること
- 著作物の意に反する改変がなされていないこと
- 引用する著作物の出所が明示されていること