

マルチブラウザのためのWebコンテンツ管理 支援システムとその応用

An Implementation of Content Translation System for Overcoming the difference in Web Browser Rendering Engines

中村正人*1 辻野友孝*1 大園忠親*1 新谷虎松*1
Masato Nakamura Tomotaka Tsujino Tadachika Ozono Toramatsu Shintani

*1名古屋工業大学大学院工学研究科情報工学専攻

Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

This paper describes the implementation of a new Web contents management system. The aim of this research is to overcome the difference in Web browser rendering engines. An advantage of our system is a functionality of imaging a part of Web pages and PC contents, such as PDF, MS office files, by using WPCScript. The WPCScript is a programming language to control the imaging function of our system. We will show how fast and flexibly the system behaves.

1. はじめに

近年、携帯電話やモバイルパソコンなどのモバイル端末が広く普及し、モバイル端末からのインターネット接続が増えている。しかし、モバイル端末は、処理能力、ネットワーク帯域がパソコンと比べ劣るため、Web ページの表示に時間がかかる。また、パソコン用の Web ページをモバイル端末で表示する場合、モバイル端末のディスプレイサイズや Web ブラウザの違いにより Web ページのレイアウトが崩れてしまう。これらの問題により、表示速度、閲覧性、操作性の点でユーザがモバイル端末上でパソコン用の Web ページを快適に閲覧することは難しい。

既存の研究では、小さなディスプレイにおけるパソコン用の Web ページの閲覧性を向上させるために、Web ページ全体をモバイル端末向けに再構築し、ユーザに配信する方法が一般的である [Kimura 06, Garofalakis 07]。しかし、これらの研究は Web ページ製作者が意図した Web ページのレイアウトを崩してしまう。閲覧性を保つために環境に依存しないレンダリング技術が必要である。

これらの課題に対処するために本研究では Web ページ画像化による Web ページ変換システムを構築する。画像にリンク機能などを実現することで Web ページとして動作するコンテンツを提供する。Web ページの画像化はレイアウトを維持できるだけでなく、Web ページのコンテンツの保護ができるという意義もある。本システムにより部分的に Web ページを画像に変換することで、コピーの防止、画像の URL 隠蔽が可能となる。また、Web ページ変換制御言語:WPCScript を実装することで Web ページ製作者が Web ページの変換を制御できる仕組みを提案する。既存の自動変換でシステムのルールに沿って変換される方式のシステムは、レイアウトが崩れたり、情報が欠落する場合があります。Web ページ製作者にとって扱いにくいものである。本研究では、Web ページ製作者が目的に合わせて Web ページの変換を WPCScript により制御する。

また、Web ページの変換を行ってもユーザが高速に表示できるシステムを構築する。同様の研究として、Chen らの研究

[Chen 03] や置田らの研究 [置田 06] が挙げられる。これらの研究は、本研究と同様サーバ上で Web サイトのレンダリングを行い、画像化したコンテンツを携帯電話に最適化して配信する手法を用いている。画像化を行う手法は、携帯電話専用のブラウザのみならずフルブラウザに対しても有益である。

2. Web ページ変換

本研究では、サーバ上で Web ページを画像に変換し、リンク機能などを持つコンテンツに変換してモバイル端末に配信するシステムを構築する。本システムが配信するコンテンツを画像化コンテンツと呼ぶ。本システムを利用することで、レイアウトを維持した Web ページ配信、Web ページ表示の高速化が可能である。また、Web ページ変換制御言語:WPCScript を実装することで Web ページ製作者が Web ページの変換を制御できる仕組みを提案する。

2.1 レイアウトを維持した Web ページ配信

Web ページを表示する場合、Web ブラウザの種類やウィンドウサイズ、モバイル端末のディスプレイサイズに依存してレイアウトが崩れる場合がある。Web ページのレイアウトが崩れてしまうと、文章が読みづらくなるもしくは目的のコンテンツが探しにくくなり、結果的にユーザビリティの低減に繋がる可能性がある。

本研究では、サーバ上で Web ページのレンダリングを行うサーバサイドレンダリングによる Web ページ画像変換技術を利用する。サーバ上の仮想的な Web ブラウザで Web ページをレンダリングすることで、Web ページのレンダリング方法の統一を図る。画像ファイルはどの端末でも表示結果が変わらない環境に依存しないフォーマットであり、Web ページのレンダリング結果を画像ファイルに変換することで端末にも Web ブラウザにも依存せずに Web ページのレイアウトを決定することができる。

2.2 高速な Web ページ表示

本システムでは、Web ページを高速に表示するために、サーバサイドレンダリング、Web ページの分割および部分的配信、画像化コンテンツの再利用を行う。ここで、サーバレンダリングは、モバイル端末に配信する Web ページをサーバ上でレンダリングする。モバイル端末上で行う Web ページのレンダリング処理を処理能力が高いサーバ上で行うことで、モバイル端

連絡先: 中村正人, 名古屋工業大学大学院 情報工学専攻 新谷研究室, 〒466-8555 名古屋市 昭和区 御器所町 名古屋工業大学, TEL:(052)735-7968, FAX:(052)735-5477, masato@toralab.ics.nitech.ac.jp

末側のリソースを低減し、処理時間を短縮する。また、サーバ上で画像化コンテンツのデータ量を削減することでロード時間を短縮する。

本システムでは、サーバからモバイル端末へ配信する画像化コンテンツを分割し、ユーザのスクロールに合わせて画像化コンテンツの部分的配信を行う。配信する画像化コンテンツを分割して必要な部分だけ配信することで、画像化コンテンツの容量を削減し、ロード時間を短縮する。そして、ユーザのスクロールに合わせて必要な部分だけ配信することで最適な画像化コンテンツ配信を実現する。

本システムでは、作成した画像化コンテンツをサーバ上に保存し、次のアクセス時にキャッシュとして再利用する。画像化コンテンツの作成は本システムで最も時間的コストがかかる処理であり、前回のアクセス時から Web ページが更新されていない場合、何度も同じ画像化コンテンツを作成するのは無駄である。そこで、本研究は画像化コンテンツを再利用することで処理の重複を避け、コンテンツ配信の高速化を図る。また、バックグラウンドで画像化コンテンツの更新確認および更新処理を行うことでデータベースには最新の画像化コンテンツが保存される。

3. Web ページ変換制御言語:WPCScript

本研究では、Web ページ製作者が Web ページの変換を制御する Web ページ変換制御言語 WPCScript を提案する。Web ページ製作者は、Web ページの変換する領域と変換方法を WPCScript によって記述することができる。

Web ページ製作者は変換する範囲を class 属性に WPCScript を設定した div タグで囲むことで画像化コンテンツに変換する範囲を記述できる。Web ページの画像化はレイアウトを維持できるだけでなく、Web ページのコンテンツの保護ができるという意義もある。現在、ブログ記事や画像などのコピーによる不正使用が問題となっている。本システムによって部分的に Web ページを画像に変換することで、コピーの防止、画像の URL 隠蔽が可能となる。

また、その div タグの中にコメントとして変換方法を記述することで、Web ページの変換方法を制御することができる。サーバ上でコンテンツ画像化と DOM 情報取得機能によって、PowerPoint などのオフィスファイルや PDF ファイルの変換、Web ページレイアウトの変更ができ、これらの機能を Web ページ上で表現することができる。図 3 はオフィスファイル変換の記述例であり、幅 300px 高さ 300px の領域で args で指定したオフィスファイルの変換を行うという意味である。本システムを利用しない場合は何も表示されないが、本システムを利用した場合変換済みのオフィスファイルが表示される。オフィスファイルの変換処理では、サーバ上で仮想的にオフィスファイルを開き、各ページ毎に画像を生成して保存する。その後、生成した画像をクリックすることで次のページへ移動できるような画像化コンテンツを生成し、Web ページ上に表示する。

3.1 出力形式

画像化コンテンツは、Web ページ画像、Web ページの機能として必要なリンクやフォーム、Flash などを記述した URL、そして画像化コンテンツを管理するためのプログラムで構成されている。メインコンテンツである Web ページ画像、入力が必要となるフォーム、コンテンツの表示が必要な Flash はそれぞれ img タグ、form タグ、embed タグを埋め込み絶対座標で表示位置を指定することで表現する。また、これらのユー

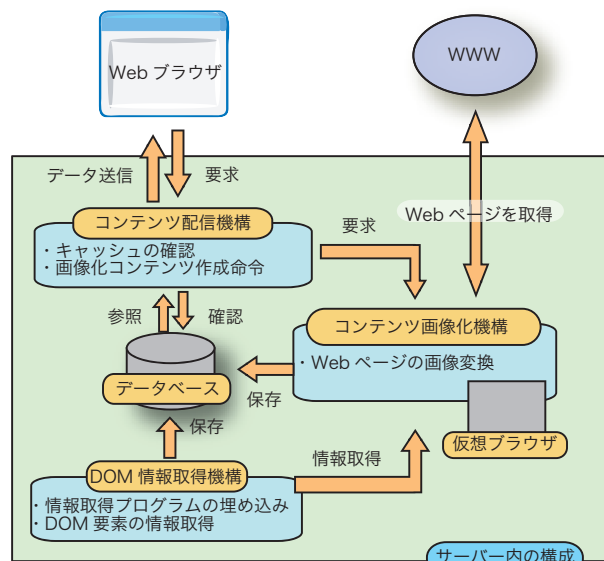


図 1: システム構成図

```

1: 入力: HTML
2: 出力: 画像化コンテンツ
3: procedure wpcs (HTML)
4: begin
5:   HTMLObject ← rendering(HTML)
6:   DIV ← {div|HTMLObject 内の div タグを示すオブジェクト}
7:   foreach div in DIV do
8:     if div.class == "WPCScript" then
9:       <Width,Height,Type,Args> ← getValue(div)
10:      Contents ← makeContent(Width,Height,Type,Args)
11:      HTMLObject ← replace(HTMLObject,Contents)
12:   return HTMLObject
13: end.
```

図 2: コンテンツ変換アルゴリズム

ザインタラクティブな要素は画像の上位に表示される必要があるため、画像の上のレイヤーにリンク、フォーム、Flashなどを配置する。このときリンクのテキスト情報は画像で表示するため、画像の上のレイヤーに透明な領域をリンクとして配置する。フォームのテキスト部分はリンクと同様に画像で表示するため、ボタンや入力フォームなどの操作に関する入力コンポーネントだけを画像の上のレイヤーに配置する。Embed タグを用いる Flash などは、そのまま画像の上のレイヤーに配置する。

3.2 WPCScript 評価器

WPCScript の処理を示す。まず、Web ページをレンダリングし、その結果から WPCScript が記述されている div タグを抽出する。そして、div タグのコメントから width, height, type, args を抜き出し、その情報に従い画像化コンテンツを作成する。div タグに上記の変数が設定されていない場合、DOM 要素から取得する。div タグ内の要素が HTML の要素であるとき、div タグが内包するコンテンツのうち、A タグ、Form を構成する要素、Embed タグの各要素のサイズや位置情報などを取得する。最後に WPCScript が記述されている div タグを画像化コンテンツと置き換える。

本システムでは、Web ページのレンダリング結果に対して DOM 情報取得プログラムを実行することで、レンダリングを試行しなければ取得できない位置情報などを取得する。このように DOM 要素を操作することで、レイアウトを変更したり、DOM 要素の情報を得ることができる。本システムでは、リンク情報の取得、WPCScript の解析、レイアウトの変更に

```
<div class="WPCScript">
  <!--
    width="300px";
    height="300px";
    type="File";
    args="./sample.ppt";
  -->
</div>
```

図 3: WPCScript の記述例

	Google	Yahoo!	Infoseek ニュース
表示領域 (高さ)	600 px	2400 px	4800 px
元の Web ページ	ファイル数	5	32
	総容量	57 KB	341 KB
	タグ数	138	434
画像化 コンテンツ	ファイル数	3	6
	総容量	63 KB	270 KB
	タグ数	5	8

表 1: Web ページと画像化コンテンツの比較

この機能を利用する。

本システムのシステム構成図を図 1 に、コンテンツ変換アルゴリズムを図 2 に示す。

本システムは、コンテンツ配信機構、コンテンツ画像化機構、DOM 情報取得機構、仮想 Web ブラウザ、キャッシュ保持のためのデータベースから構成される。端末からのアクセスがあると、本システムはまずデータベースにより画像化コンテンツのキャッシュの有無を確認する。既にデータベースに画像化コンテンツがあれば、そのコンテンツを配信した後コンテンツの更新チェックを行う。そこでコンテンツが更新されていれば、画像化コンテンツを新たに作成する。また、データベースに元々画像化コンテンツが存在しなければ、画像化コンテンツを新たに作成し配信する。画像化コンテンツは、サーバ上の仮想 Web ブラウザ上で Web ページをレンダリングして生成する。このとき、DOM 情報取得機構により、画像化するコンテンツ及びコンテンツの内包するコンテンツの位置情報、サイズ情報などを取得する。また、WPCScript の解釈を行い、レイアウトの変更を適用する。そこで得られたレンダリング結果を画像として保存する。WPCScript においてドキュメントファイルが指定された場合は、ファイルの変換処理を行う。生成した画像及びユーザインタラクティブな要素に関する情報を基に、コンテンツ配信機構によりコンテンツを生成し、クライアントへ配信する。

4. 評価

モバイル端末のフルブラウザで元の Web ページを表示した場合と変換した画像化コンテンツを表示した場合の表示速度を比較する。モバイル端末は Apple 社の iPod touch と 3G 携帯電話の SoftBank 911T を使用した。iPod touch のブラウザである Safari 上、携帯電話の PC サイトブラウザ上でそれぞれ元の Web ページとサイト全体を変換した画像化コンテンツを表示する。サーバは CPU 2.66GHz Dual-Core Intel Xeon, メモリ 2GB の Mac OS X 10.4 を使用した。

表 1 に Web ページと画像化コンテンツの比較を示す。Web ページの表示時間はリクエスト開始から Web ページ全体の表示が完了するまでの時間を計測し、表 2 に携帯電話、表 3 に iPod touch で各 Web ページを表示した時間の比較を示す。

まず、画像化コンテンツを作成後に配信した場合の表示速度について考察を行う。表 1,2,3 を見ると Google は表示速度

	Google	Yahoo!	Infoseek ニュース
Safari	1.91s	6.05s	13.48s
画像化コンテンツを作成後に表示	2.98s	5.93s	9.38s
画像化コンテンツを先に表示	1.89s	2.32s	3.53s

表 2: 携帯電話上での表示時間の比較

	Google	Yahoo!	Infoseek ニュース
PC サイトブラウザ	5.05s	14.6s	25.83s
画像化コンテンツを作成後に表示	5.17s	10.02s	19.35s
画像化コンテンツを先に表示	1.48s	6.84s	13.13s

表 3: iPod touch 上での表示時間の比較

が遅くなっているが Yahoo! と Infoseek ニュースでは向上したことが分かる。Google は変換後の総容量が大して変化しなかったことと表示領域が狭いことが要因となり Web ページを変換した時間が表示速度に影響し表示速度が遅くなっている。Yahoo! や Infoseek ニュースのような表示領域が広く、コンテンツが多い Web ページでは、変換後の総容量、ファイル数、タグ数が削減されているため表示速度が向上している。

5. まとめ

モバイル端末における Web ブラウジングでは、レイアウトが崩れる、表示速度が遅いといった問題がある。レイアウトが崩れるという問題は、モバイル端末に限らず発生する問題である。その問題に対し、Web ページの画像化によるレイアウトを保った Web ページ配信を行った。Web ページ製作者が Web ページ変換を制御できる WPCScript を利用することで、Web ページの部分的なコンテンツ変換ができる。画像化がもたらす付加的な利点として、コピーの防止、画像の URL 隠蔽が挙げられる。

また、表示速度が遅いという問題に対し、サーバレンダリング方式、Web ページの分割および部分的配信、画像化コンテンツの再利用を行った。表示速度に関して比較実験を行い、Web ページの変換を行っても表示速度の向上が見られたため、表示速度において本研究は有益である。

さらに、コンテンツ変換の分野において、既存の研究では Web ページ製作者の意図を機械的に解釈するものが多い。本研究では Web ページ製作者の意図を Web ページ製作者が Web ページ変換制御言語 WPCScript によって記述でき、半自動的な変換手法となっている。従来の Web ページ変換よりも柔軟でより Web ページ製作者の意図が反映できる点で有益である。

参考文献

[Kimura 06] A. F. R. Rahman, H. Alam, R. Hartono and K. Ariyoshi : "Automatic Summarization of Web Content to Smaller Display Devices," in Post Presentations of 6th International Conference on Document Analysis and Recognition, Sept. 10-13, 2001.

[Garofalakis 07] J. Garofalakis, V. Stefanis : "Using RSS feeds for effective mobile web browsing", in Proc. Information Society Journal, Volume 6, Number3/November, pp.239-257, 2007.

[Chen 03] Yu Chen, Wei-Ying Ma, and Hong-Jiang Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In WWW '03: Proceedings of the 12th international conference on World Wide Web, pages 225-233, New York, NY, USA, 2003. ACM.

[置田 06] 置田 誠, 山口 典男, 重松 隆之, 高橋 修, and 宮本 衛市. 携帯電話機能 web ブラウザのサーバ・レンダリング方式の提案と実装評価 (モバイルアプリケーション. [特集] マルチメディア, 分散, 協調とモバイル (dicomo2005)). 情報処理学会論文誌, 47(7):2107-2116, 20060715.