

マルチブラウザのための Web コンテンツの自動変換環境とその応用

近藤 圭佑[†] 浅見 昌平[†] 大園 忠親[†] 新谷 虎松[†]名古屋工業大学大学院工学研究科情報工学専攻[†]

1 はじめに

近年、様々な Web ブラウザが登場し、ユーザは機能などから自分に合う Web ブラウザを選択するようになった。また、モバイル端末上でもフルブラウザによってパソコン用の Web ページを表示できるようになっている。

しかし、Web ブラウザはそれぞれ異なるレンダリングエンジンを使用していることもあり、どの Web ブラウザでも同じように表示することは難しい状況である。また、レンダリングエンジンの違いだけではなく、端末や Web ブラウザのウィンドウサイズが異なる場合、Web ページのレイアウトが崩れてしまうという問題もある。そのため、どのような環境でも一様なレイアウトで Web ページを表示することは難しい。

そこで本研究では、どのようなブラウザでも一様なレイアウトが表示できる環境を整えることを目的とし、マルチブラウザのための Web コンテンツの自動変換環境を構築する。マルチブラウザのための環境とは、どの Web ブラウザでも一様に Web ページを表示できる環境である。本システムは、Web ページを指定するとサーバ上で Web ページを仮想的にレンダリングして画像に自動変換する。サーバ上で単一の設定でレンダリングを行っているため、Web ブラウザに依存せずにレイアウトを保って表示することができる。また、サーバ上で Web コンテンツを自動変換する技術を利用して本研究で提案する WPCScript により、Office ファイルの変換や、レイアウトの指定、Javascript の実行といったレイアウトの変更ができる。以降、本システムが変換して配信するコンテンツを画像化コンテンツと呼ぶ。本システムが作成した画像化コンテンツは応用としてモバイル端末に配信することも可能である。

2 関連研究

Web ページを自動変換する研究は主にディスプレイサイズが制限される環境で表示するために行われてきた。文献 [1] は 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する。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。文献 [2] では、Web ページの構造や内容を解析して、小さい画面のデバイスのために、要約した Web ページを表示するシステムを提案している。文献 [3] では、Web ページの要素に対して、Google のページランクに似たランク付けアルゴリズムを利用することで、小さい画面のデバイスに対しての Web ページを生成している。具体的には、元となる Web ページを要素化して、ランク付けアルゴリズムを使い、要素をツリー状に構造化させる。文献 [4] は RSS に着目して Web ページのガイドラインをつくりコンテンツに誘導するものである。しかし、これらはテキストや画像を配信するものでレイアウト情報などは保たれていないので Web ページの製作者が意図するものと異なってしまう。本研究では Web ページの内容だけではな

くレイアウトも重要な情報だと考えているためこれらの研究は不十分であると考えた。

3 Web コンテンツ自動変換システム

3.1 レイアウトを維持した Web ページ配信

本研究では、Web ブラウザの種類に依存せず Web ページ製作者が意図したレイアウトをそのまま表示できる環境を目標としている。しかし、通常 HTML を表示する場合、使用する Web ブラウザに依存して多少レイアウトが変化する。また、同じ Web ブラウザでもディスプレイサイズやウィンドウサイズなど環境によってもレイアウトは変化する。この問題を解決する方法は、自前でレンダリングエンジンを開発しすべてのマシンに載せる方法、単一の Web ブラウザでレンダリングを行いその結果をモバイル端末に送る方法があり、本研究では後者の方法を用いた。

本研究では、サーバ上に仮想的な Web ブラウザを用意し、その仮想 Web ブラウザ上で Web ページのレンダリングを行う。そして、その Web ページのレンダリング結果を画像に変換して配信する。jpeg や png などの画像はどの端末でも同様に表示される環境に依存しないフォーマットである。また、サーバ上で単一の設定でレンダリングを行っているため、どの端末にもどの Web ブラウザにも依存せずにレイアウトを保って表示することができる。

しかし、Web ページを画像化して配信するだけではリンクやフォーム、Flash、Javascript などの機能を表現することができない。そこで、本システムは Web ページ画像とリンク、フォームなどの機能だけを残した HTML をひとつのコンテンツとして既存の Web ブラウザ上に配信する。この機能を実現するために、本システムは、サーバ上で Web ページをレンダリングした結果から DOM 情報を取得する。レンダリングした結果から取得したリンクやフォーム、Flash の座標や URL の情報を HTML 形式で Web ページの画像とともに配信することで画像に Web ページとしての機能を持たせる。Web ページを画像化してしまうと Javascript を利用した動的に変化する Web ページに対応できない。そこで、ユーザが Javascript を動かす場合、本システムが Web ブラウザ上で動作することを利用して、本システムは対象の Web ページのリンク先を変更して配信する。変更されたリンクは本システムで変換した画像化コンテンツの URL である。リンク先を画像化コンテンツの URL に変更することで、配信した Web ページは本システムの一部として動作することができる。

3.2 DOM 情報取得

ここでは画像化コンテンツにおいて、リンク、フォーム、Flash を画像上で実現するためにこれらの情報を抽出する DOM 情報取得機能について述べる。

DOM 情報取得機能は、サーバ上で Web ページをレンダリングする際に、Web ページに DOM を操作するプログラムを付加してそのレンダリング結果に対して実行する機能である。この機能によって、Web ページの DOM 情報を操作してレイアウトを変更したり、ある DOM 要素の情報を得ることができる。この機能は HTML を解析するだけでは得られないような DOM 要素の位置情報を得ることができる。本システムではこの機能を使ってリンクなどの位置情報を抽出している。他にも、フォントサイズを変更したり、配置を変更することもできる。

A Web Content Converter for a Multi Browser and Its Applications

Keisuke KONDO, Shohei ASAMI, Tadachika OZONO, and Toramatsu SHINTANI

[†]Dept. of Computer Science and Engineering, Graduate School of Engineering Nagoya Institute of Technology, 466-8555, Nagoya, JAPAN

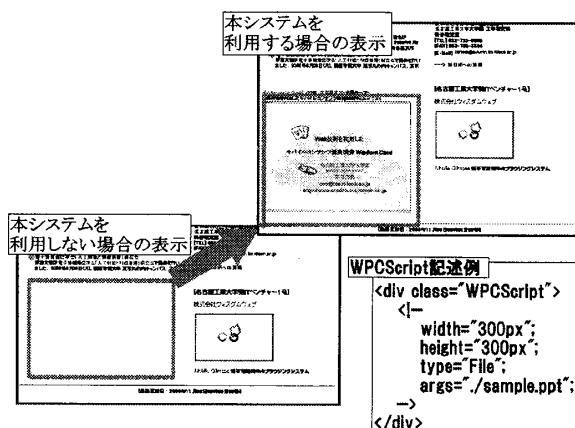


図 1: WPCScript の記述例と実行例

3.3 WPCScript

Web ページ製作者が Web ページを製作するときに、本システムを利用することで通常の Web ページとは違った表現が可能となる。通常の Web ページと違った表現を実現するために、Web ページの部分的画像化コンテンツ変換、WPCScript という機能を本システムでは提供している。この機能は、Web ページ製作者が本システムが提供する簡単なスクリプト WPCScript を記述することで利用可能となる。本システムは、まず DOM 情報取得機能によって Web ページを解析し、WPCScript が含まれているタグを発見し、その位置情報と WPCScript を抽出する。抽出した WPCScript を解析し、Web ページの WPCScript を含んでいたタグの部分を変換する。この機能を使用することで、Web ページの一部を画像化コンテンツに変換することができ、コンテンツの保護をすることもできる。

WPCScript には、3つのタイプがあり、Office ファイルの変換、レイアウトの選択、レイアウトの記述ができる。Office ファイルの変換では、Web ページの一部にファイルを指定するスクリプトを記述するだけでページめくりができる Office ファイルコンテンツを Web ページに組み込むことができる。レイアウトの選択、記述はレイアウトを変更するためのスクリプトを Web ページに付加することで、本システムは Web ページをレンダリングする際に Web ページの DOM 情報を操作してから画像化することができる。これは、サーバ上でのコンテンツ画像化技術と DOM 情報取得技術を応用することで実現可能な応用例である。

図 1 に Office ファイルの変換の記述例と実行例を示す。WPCScript は class 属性を “WPCScript” とした div タグ内にコメントとして記述する。図 1 右下の記述例は幅 300px 高さ 300px の領域で args で指定した Office ファイルの変換を行うという意味である。Type は処理の種類であり、File (Office ファイル変換)、Layout (レイアウト選択)、Script (レイアウト記述) が指定できる。args には引数を記述する。実行例では、本システムを利用しなかった場合、枠で囲まれた部分は表示されていないが、本システムを利用した場合 Office ファイルが表示されていることがわかる。Office ファイルをクリックすることでページ変更ができるようになっている。

4 考察

Web ページの閲覧においてユーザが重要視することは表示速度、閲覧性、操作性の3点だと本研究では考える。ここでは、表示速度は、Web ページにアクセスしてからロードが完了しレンダリングが終了するまでの時間、閲覧性とは Web ページを表示したときの内容を読みやすさ、操作性はユーザの操作する頻度や操作しやすさとする。本研究では Web ページを画像化することでレイアウトを保ち、ブラウザに依

表 1: 画像形式によるファイルサイズの比較

形式	ファイルサイズ [KByte]	
	文字画像	芝の画像
jpeg 100%	0.83	1.32
jpeg 70%	0.54	0.46
png 24	0.53	1.2
png 8	0.4	0.89

存しない環境を構築している。しかし、画像の容量をある程度減らさなければ表示速度が遅くなってしまふ。画像の容量を減らしすぎると閲覧性が落ちてしまふ。本章では、どのような形式で Web ページを画像化すればいいかを考察する。

本システムでは文字を画像に変換する。文字はノイズが入ると視認性が悪くなる。jpeg 形式の場合、圧縮を行うと劣化するため、画像の形式は jpeg 形式ではなく png 形式を用いたほうがいいと考えられる。

しかし、グラデーションや写真のような画像の場合は jpeg 形式のほうが適している。画像は文字と比べてノイズが入っても視認性にあまり問題がなく、ファイルサイズを小さくできる jpeg 形式のほうがよいと考えられる。

jpeg と png でファイルサイズを比較した場合の結果を表 1 に示す。これは、20x20 ピクセルの白い背景に、文字を表示させた場合と、芝の画像を表示した場合のファイルサイズである。png8 は 256 色、png24 は 1679 万色を表現可能な形式である。また、表中の文字の場合は、ファイルサイズの面から見ても、png の方が適していることが確認できた。また、文字は表現すべき色数が少ない場合が多い。そのため、文字を変換して画像化する際は、png8 を用いるほうがよい。

5 まとめ

本研究では、どのようなブラウザでも一般的なレイアウトが表示できる環境を整えることを目的とし、マルチブラウザのための Web コンテンツの自動変換環境を構築した。本システムは、Web ページをサーバ上で画像化し、Web ページとしてリンクなどの機能を付加した画像化コンテンツに変換する。サーバ上で単一の設定でレンダリングを行っているため、Web ブラウザに依存せずにレイアウトを保って表示することができる。また、ディスプレイサイズが小さいモバイル端末上でもレイアウトを保ったまま表示ができる。

また、本研究で提案する WPCScript を利用することで Web ページ製作者は、指定したコンテンツだけを変換可能となる。WPCScript は、Office ファイルの組み込み、レイアウト指定、Javascript の実行といったレイアウトの変更方法を記述できる。ユーザは本システムを通して Web ページにアクセスすることで、システムは WPCScript を解釈し、変換して配信する。

参考文献

- [1] O. Buyukkokten, H. Garcia-Molina, A. Paepcke, and T. Winograd: “Power browser: Efficient web browsing for PDAs”, in Proc. Human-Computer Interaction Conference 2000 (CHI 2000), vol. 2, issue 1, pp. 430-437, Apr, 2000.
- [2] 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, Seattle, The United States, Sept. 10-13, 2001.
- [3] Xinyi Yin, Wee Sun Lee: “Using Link Analysis to Improve Layout on Mobile Devices”, in Proc. of World Wide Web Conference (WWW '04), New York, May, 2004.
- [4] John Garofalakis, Vassilios Stefanis: “Using RSS feeds for effective mobile web browsing”, in Proc. Information Society Journal, Volume 6, Number3/Novemmer, pp.239-257, 2007.