

動的コンテンツに利用可能な日本語 Web フォント配信システム

奥村 命† 案西 稿志†† 永井 明彦†††† 伊藤 孝行†, †††, ††††

†名古屋工業大学 産業戦略工学専攻 ††株式会社モリスワ †††東京大学政策ビジョン研究センター
†††名古屋工業大学 グリーン・コンピューティング研究所

1 はじめに

WWW の発展と共に、Web ブラウザも大きな進化を遂げてきた。文字装飾、画像、動画等の文章以外のリッチなコンテンツを扱えるようになり、更に Ajax, Flash などの高度な表現技術が登場している。しかし、Web サイト運営者が意図するフォントを利用するには、フォント適用後のテキストを画像化することで代用されている。本問題の解決方法として、CSS3による Web フォント技術 [1] が登場し、本技術を利用したフォント配信システムが提案されている。例えば、Google Web Fonts[2] である。図 1 は、Google Web Fonts[2] のアクセス数である。図 1 から、順調にアクセス数が伸びており、フォント配信システムが求められていることが分かる。

しかし、Google Web Fonts[2] を含め、既存の Web フォント配信システム [2, 3, 4, 5] は、ページ遷移無しにコンテンツが動的に変化するような Web サイトの増加、モバイル端末からの利用増加等の Web サービスの現状を十分考慮できていない。本論文では、本課題を解決するフォント配信システムを提案する。

2 システム要件

[課題 1] Flash を利用しない実装

論文 [6] では、Flash によるフォント配信システムを実現している。一方、近年ではスマートフォン、タブレット端末等のモバイル端末が普及してきており、モバイル端末上への対応が望まれる。モバイル端末向けの Flash Player 提供の打ち切り、iOS 端末での Flash 非サポート等の理由から、Flash を利用した実装は問題である。従って、本論文で提案するシステムでは、CSS3 と JavaScript によって実装している。

[課題 2] フォントのサブセット化

モバイル端末での利用を想定し、低速ネットワーク、低処理能力端末上で動作するシステムでなければならない。アルファベット圏のフォントであれば、特殊文字を合わせても 100 文字程度しかないため、容易にフォント全体を送信することが可能である。しかし、日本語フォントの場合は 1 つの書体で、フォントのファイルサイズが 5~10M byte もの大きさになり、フォント全体を送信することは難しい。実際に、モバイル端末で日本語フォント全体を受信すると、ブラウザアプリケーションが強制終了してしまう場合がある。従って、本システムでは、フォント全体から”必要な部分”のみを抽出して配信する。本論文では、フォントから”必要な部分”のみを抽出することを、フォントのサブセット化と定義する。既存のフォント配信システム [2, 3] は、

Google Web Font Growth

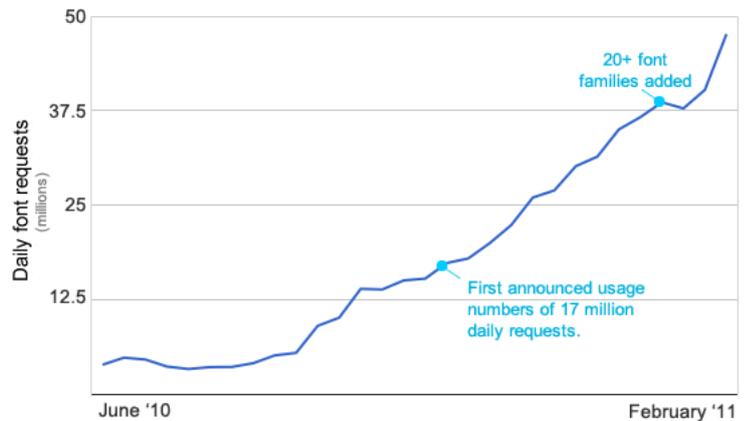


図 1: Google Web Fonts のアクセス数
”<http://googlewebfonts.blogspot.com/2011/02/google-web-fonts-new-stats-new-name.html>”

フォントのサブセット化を行っておらず、ファイルサイズの大きな日本語フォントに対応していない。

[課題 3] 動的コンテンツへの対応

近年の Web サイトでは、ページ遷移を伴わず、コンテンツが動的に変化する場合がある。描画されている文字が変化することにより、フォントの必要とされるサブセットが変化するため、動的にフォントを生成、適用し直さなければならない。フォント配信システム [4, 5] は、フォントのサブセット化を行っているが、ページ遷移を伴わないコンテンツの動的変化に対応することができない。

以上の課題を解決するために、本論文では、コンテンツの動的変化に対応して、”必要な時”に”必要な部分”のフォントをサブセット化して配信する動的フォント配信システムを提案する。

3 動的フォント配信システム

具体的なシステムプロセスを図 2 に示し、詳細な処理ステップを下記にまとめる。

[STEP 1] フォント適用の設定

[METHOD 1]~[METHOD 3] で説明する方法でフォント適用先、適用フォント等を設定する。設定項目の詳細は [METHOD 2] にあげた。

[STEP 2] フォント生成リクエスト

ページのロード時、[METHOD 2] で指定したイベントが発生した時、及び [METHOD 3] で直接フォントを要求した際に、本ステップが実行される。適用フォント、フォントタイプ (SVG, WOFF, TrueType)、適用先に含まれる文字列をサーバー側に送信し、フォント

†Mikoto OKUMURA ††Kouji ANZAI ††††Akihiko NAGAI †, †††, ††††Takayuki ITO
†Master of Techno-Business Administration, Nagoya Institute of Technology
††Morisawa Inc.
†††Todai Policy Alternatives Research Institute
††††Center for GREEN COMPUTING, Nagoya Institute of Technology

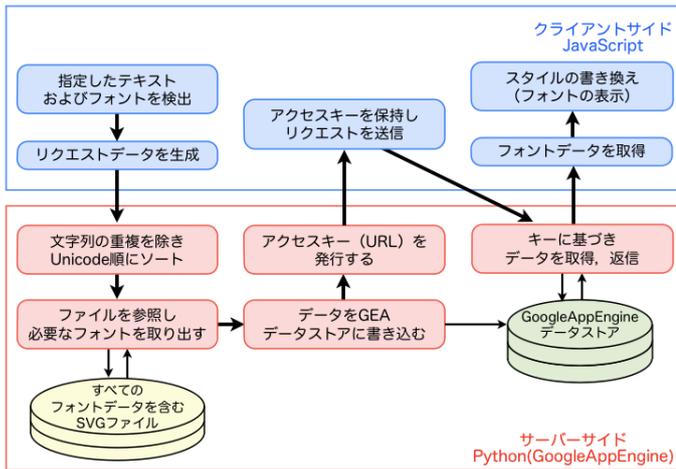


図 2: システムプロセス図

をリクエストする。

[STEP 3] フォントのサブセット化

[STEP 2] により受け取った文字列を Unicode 順にソートし、重複を取り除く。SVG 形式で保存されているフォントデータから、必要な部分のみを切り取り、サブセット化されたフォントを作成する。

[STEP 4] サブセット化したフォントへのキーを割当

[STEP 3] で生成されたフォントを保存し、一意のキーを割り当てる。キーをクライアント側に、レスポンスとして返却する。

[STEP 5] フォントデータリクエスト

[STEP 4] で得たキーをもとに、対応するフォントデータを要求する。[STEP 4] でキーを作成した後、フォントデータを要求する理由は、下記に示す 4 つの制約があり、直接フォントを URL で指定できないため、キーを用いた短縮 URL を用いてフォントを指定する必要があるからである。

- 制約 1. CSS3[1] の仕様上、フォントを指定する際に URL で指定する必要がある。
- 制約 2. [STEP 2] で示したパラメータを指定するために、URL にパラメータを含めなければならない。
- 制約 3. ブラウザ毎に URL 長の制限がある。
- 制約 4. サブセット化する部分を指定するパラメータが長くなる。

本論文で提案する動的フォント配信システムは、JavaScript のライブラリとして整備されており、利用者の技術レベルに応じて下記 3 つの利用方法を想定する。

[METHOD 1] 命名規則に則った class 属性による設定
下記の命名規則に則った class 属性を、フォントを適用したい HTML タグに付与することで、指定のフォントを適用することができる。

[Identifier]_[FontName]_swf
[Identifier] : 識別子 (自由)
[FontName] : 適用フォント名

本指定方法は、最も簡単な指定方法で、利用者にプログラミング知識を要求しない。ただし、本指定方法では、コンテンツの動的な変更に対して対応しない。

[METHOD 2] JSON による設定

下記に設定方法を示す。

```
Dwf.set({
  font      : font,
  fontType  : fontType,
  target    : {className: "targetClass"},
  events    : ["event1", "event2", "event3"],
  listener  : {id: "listenerId"},
  handler   : function() {...}
});
```

font: 適用フォント名
fontType: 適用フォントタイプ
target: フォントの適用先
events: 発生時にフォントを適用し直すイベント
listener: イベントのリスナー
handler: イベント検知時に実行される関数

target 及び listener は、JSON オブジェクトで指定し、プロパティとして id, className, 及び elements を利用できる。順に、指定要素の id, クラス名, HTML 要素を指定する。3 つのプロパティは同時に利用可能で、指定された HTML 要素全てを対象とする。

本指定方法では、表示テキストの変更、ボタンのクリック、一定時間経過等のイベント発生に応じて、適応するフォントを生成、適用し直すことが可能である。本機能により、ページの遷移を伴わないコンテンツの動的変化に対応したフォント配信システムを実現する。また、JSON 形式による宣言的な記法により、高度なプログラミング技術を要求せずに、動的なフォント配信システムを利用することが可能である。

[METHOD 3] JavaScript によるプログラミング

詳細な制御を必要とする場合、FontManager クラスを利用することで、JavaScript による制御を可能とする。FontManager クラスは、適応するフォントの切り替え、サーバーへの要求、HTML 要素への適用等を制御するメソッドを持つ。

4 まとめ

本論文では、“必要な時”に“必要な部分”のフォントを配信することが可能な動的フォント配信システムを提案した。“必要な時”に配信することで、コンテンツが動的に変化する Web サイトに対応できる。フォントを“必要な部分”のみにサブセット化して配信することで、日本語フォントのようなファイルサイズの大きなフォントを配信することを可能にしたとともに、モバイル端末での利用を想定した低速ネットワーク、低処理能力端末環境下でも動作が可能となる。

謝辞

本システムを実装するにあたり、名古屋工業大学 伊藤孝行研究室を 2011 年 3 月に卒業された高橋侑也氏には、サーバーサイドのプログラム作成で多大な協力を得た。ここに感謝の意を表す。

参考文献

- [1] W33C CSS Fonts Module Level 3, <http://www.w3.org/TR/css3-fonts/>
- [2] Google Web Fonts, <http://www.google.com/webfonts>
- [3] Typekit, <https://typekit.com/>
- [4] デコもじ, <http://decomoji.jp/>
- [5] フォントプラス, <http://webfont.fontplus.jp/>
- [6] 高橋 侑也, 安藤 哲志, 伊藤 孝行, 装飾された美しい日本語フォント配信システムの提案. 電子情報通信学会総合大会講演論文集, 社団法人電子情報通信学会, 2010(2), 215, 2010-03-02