

2009年度 修士論文

携帯端末用 Web ページから通常端末用ページを
構成する手法に関する研究

2010年2月1日(月) 提出

指導：深澤良彰 教授

早稲田大学大学院 基幹理工学研究科
情報理工学専攻

学籍番号：5108B028-7

勝俣 浩樹

目次

| | | |
|-----|------------------------|----|
| 第1章 | はじめに | 1 |
| 第2章 | 携帯端末向け Web ページ | 3 |
| 2.1 | 携帯端末向け Web ページ | 3 |
| 2.2 | デバイス能力に特化した Web ページの問題 | 4 |
| 第3章 | 本手法の概要 | 6 |
| 3.1 | 本手法の提案 | 6 |
| 3.2 | 本手法の構成 | 6 |
| 第4章 | Web 文書構造モデル | 8 |
| 4.1 | 既存の携帯端末用 Web ページの分析 | 8 |
| 4.2 | Web 文書構造の定義 | 12 |
| 4.3 | Web 文書構造モデルの定義 | 14 |
| 第5章 | 通常端末向け Web ページの構成 | 15 |
| 5.1 | 前提条件 | 15 |
| 5.2 | Web 文書構造抽出とモデルへの変換 | 16 |
| 5.3 | 通常端末向け Web ページの構成 | 26 |
| 第6章 | 評価 | 30 |
| 6.1 | 評価手法 | 30 |
| 6.2 | 結果 | 30 |
| 6.3 | 考察 | 31 |
| 第7章 | 関連研究 | 32 |
| 第8章 | おわりに | 34 |
| | 謝辞 | 35 |

| | |
|--------------------------|----|
| 参考文献 | 36 |
| 研究業績 | 38 |
| 付 録 A Web 文書構造モデルのスキーマ定義 | 39 |

第1章 はじめに

今日、Web が情報媒体として広く利用されている。Web ページの閲覧には PC を利用するのが一般的であるが、近年、携帯電話をはじめとする移動用の小型携帯通信機器が PC と並んで普及している。しかし、両者の Web ページ閲覧に用いられるデバイス能力 (device capability) には差があるため、Web サイトはその能力に特化して構築されることが多い。具体的には、画面サイズに合わせたレイアウト、機器上で稼動する Web ブラウザの能力に合わせたタグ・属性・スタイルシートの制限などである。本論文では、PC を代表とする機器を「通常端末」、携帯電話をはじめとする小型携帯通信機器を「携帯端末」と呼ぶ。

携帯端末向けに特化した Web ページは、概して通常端末向け Web ページに比べて相対的に低い能力しか要求しない構成になっている。例えば、携帯端末の小型画面に合わせて Web ページのレイアウトが構成されるため、1 ページに少量のコンテンツしか含まないように設計されている。このようなデバイス能力に特化した Web ページを能力の異なるデバイスで閲覧する場合、問題が生じることがある。通常端末の点から考えると、これは携帯端末よりもデバイス能力が相対的に高いため、携帯端末向けに特化した Web ページの閲覧は可能ではある。しかし、ページあたりに少量のコンテンツしか含まない設計のために、複数ページを連続して閲覧する必要が生じ、ページ遷移におけるクライアント・サーバー間通信が頻繁に発生するなど、利用経験 (user experience) が快適であるとは言いがたい。通常端末による閲覧の場合、コンテンツが 1 ページにまとまっていることを期待するにもかかわらず、その構成は携帯端末向けに特化していることから、利便性の著しい低減を感じることになる。

我々はこの問題に対し、画面サイズの違いに起因する快適性の損失に着目した上で、携帯端末の小型画面向けに特化した Web ページから通常端末の画面向け Web ページを構成する手法を提案する。本論文では、画面サイズに依存しない Web ページの構成を抽出するために、ページの構成要素が持つ意味をモデル化し、デバイス能力に応じた Web ページに変換する手法について述べる。

本論文の構成を次に示す。2 章では、携帯端末向け Web ページの詳細を説明する。3 章では、本手法の概要について述べる。4 章では、既存の携帯端末向け Web

サイトの調査による文書構造の傾向の分析結果について述べ、それに基づいたモデルを定義する。5章では、携帯端末向け Web ページから文書構造を抽出してモデル化を行い、このモデルから通常端末向け Web ページを構成する手法について述べる。6章では、現実の携帯端末向け Web サイトを対象として、本手法の評価とその結果について述べる。

第2章 携帯端末向け Web ページ

本章では、携帯端末向け Web ページの詳細と、デバイス能力に特化した Web ページの問題について述べる。

2.1 携帯端末向け Web ページ

本研究で述べる携帯端末向け Web ページは、携帯端末のデバイス能力に特化して構築されたページを指す。これは一般的に、通常端末向け Web ページに比べて相対的に低い能力しか要求しない構成になっている。同一の Web サイトにおける通常端末向け Web ページと携帯端末向け Web ページの例を図 2.1 に示す。



図 2.1: デバイス能力に特化した Web ページの例 [1]

図 2.1 において、左が通常端末 (PC の Web ブラウザ) を用いて通常端末向け Web ページを閲覧した際の画面、右が携帯端末 (携帯電話の Web ブラウザ) を用いて携

帯端末向け Web ページを閲覧した際の画面である。両者の大きな違いとして、図 2.1 のように、画面サイズに合わせた Web ページのレイアウトが挙げられる。携帯端末の小型画面向けにレイアウトされた Web ページは通常端末向け Web ページに比べ、1 ページに少量のコンテンツしか含まないように設計されている。また、携帯端末向け Web ページの文書を表示するマークアップ言語の仕様は通信キャリアが独自に策定していることが多い [2][3][4] が、これらは概して、既存の HTML/XHTML の仕様 [5][6] よりもタグ・属性・スタイルシートの表現能力が制限されたものとなっている。

2.2 デバイス能力に特化した Web ページの問題

デバイス能力に特化した Web ページのほとんどは、能力の異なるデバイスを考慮して設計されていないため、閲覧に問題が生じることがある。ここで、通常端末から携帯端末向け Web ページを閲覧することを考える。通常端末は携帯端末よりもデバイス能力が相対的に高いため、携帯端末向けに特化した Web ページは閲覧可能である。しかし、これは一方で、携帯端末向け Web ページが相対的に低い能力しか要求しないことから、通常端末の高いデバイス能力が使用されていないということでもある。

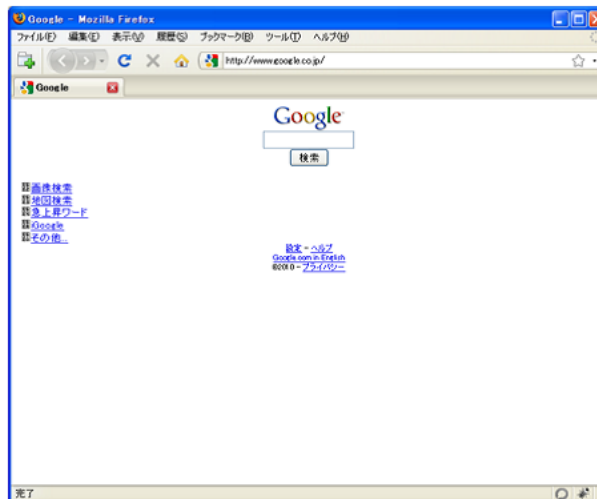


図 2.2: 通常端末で携帯端末向け Web ページを閲覧した例

図 2.2 は、図 2.1 で示した携帯端末向け Web ページを通常端末で閲覧した例である。図 2.2 のようにページあたりのコンテンツが少なくなると、必要な情報を得るために参照するページ数が増加すると考えられる。その結果、複数ページを連続して閲覧する必要が生じ、ページ遷移におけるクライアント・サーバー間通信が頻繁に発生するなど、利用経験 (user experience) の快適性を損ねる事象が発生する。これが携帯端末からの閲覧であれば、当然受け入れられるべきものであり、閲覧者も意図せずに許容する。しかし、通常端末からの閲覧の場合、コンテンツが 1 ページにまとまっていることを期待するにもかかわらず、その構成は携帯端末向けに特化していることから、利便性の著しい低減を感じることになる。同様に、タグ・属性・スタイルシートの制限による文書表現の貧弱さなども利用経験の快適性を損ねる要因となるが、本研究では特に、画面サイズの違いに起因する快適性の損失に着目している。

第3章 本手法の概要

本章では、携帯端末向け Web ページから通常端末向け Web ページを構成する手法についての概要を述べる。

3.1 本手法の提案

本研究は、デバイス能力に特化した Web ページを能力の異なるデバイスで閲覧する際の、2.2 節に述べた問題の解決を目的としている。その中で、通常端末から携帯端末向け Web ページを閲覧する際の、画面サイズの違いに起因する快適性の損失に着目した。本論文では、携帯端末の小型画面向けに特化した Web ページから通常端末の画面向け Web ページを構成する手法を提案する。具体的には、画面サイズに依存しない Web ページの構成を抽出するために、ページの構成要素が持つ意味をモデル化し、デバイス能力に応じた Web ページに変換する手法について述べる。

3.2 本手法の構成

本手法は、図 3.1 に示す構成である。図 3.1 において、入力・出力を四角で括った文字、入力の加工を括弧付きの文字で表現している。また、矢印が入力の加工と出力の順序を示しており、図の上から下の順で処理が行われる。

本手法は外部からの入力として、2.1 節に述べた携帯端末向け Web サイトを必要とし、この Web サイトの各ページから特定の意味を持つ文書構造を抽出する。本研究で述べる「文書構造」は、単純な文書としての構成だけでなく、Web における意味・役割を持つ構造と見なしていることから、この構造を「Web 文書構造 (文書構造)」と呼ぶ。

Web ページから抽出された Web 文書構造は、画面サイズに依存しない構造に変換する。本論文では、この構造を「Web 文書構造モデル (モデル)」と呼ぶ。この

モデルは特定の意味を持つ文書構造を元に定義したものであり、4章でその詳細を述べる。

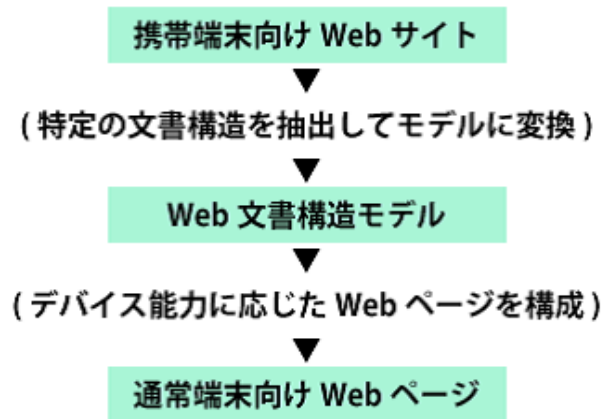


図 3.1: 本手法の構成

最後に、生成された Web 文書構造モデルからデバイス能力に応じた Web ページを構成する。本論文では、出力対象のデバイスを通常端末と想定していることから、手法適用の結果として通常端末向け Web ページが出力される。Web ページの構成手法については、5章で詳細を述べる。

第4章 Web文書構造モデル

本章では、既存の携帯端末向け Web ページの調査による Web 文書構造の分析結果と、それに基づいた Web 文書構造モデルの定義について述べる。

4.1 既存の携帯端末用 Web ページの分析

携帯端末による Web 閲覧の普及により、一般に公開されている携帯端末向け Web サイトは無数に存在する。これら Web ページの多くは、通常端末向け Web ページに比べて相対的に低い能力しか要求しない構成となっていることから、ページの装飾に費やされる構成要素が限定されている。したがって、互いに無関係な複数の Web サイト間においてもページを構成する文書構造が類似しやすいと考えられる。また、これを通常端末で閲覧する場合、ページの装飾は通常端末上で閲覧するには貧弱であり、情報としての価値も低い。これをふまえ、本研究では既存の携帯端末向け Web サイトのうち、PC から閲覧可能な約 50 サイトの HTML 文書を調査した。その内訳を表 4.1 に示す。

表 4.1: Web サイトの種類と調査数

| 構成 | 種類 | 調査数 |
|------------|-------------------|-----|
| 文字が中心のサイト | ニュースサイト、掲示板、ブログなど | 18 |
| 画像を多く含むサイト | 動画投稿サイト、地図検索サイトなど | 13 |
| 情報が多様なサイト | SNS サイト、ポータルサイトなど | 15 |

調査対象となる Web サイトは、その種類に偏りが無いように選択しているが、表 4.1 ではこれを大まかな構成ごとにグループ分けしている。調査の結果、携帯端末向け Web ページの構成要素について、以下の 3 種類に分類されると考察した。

- (1) 主体となる情報

- (2) 列挙を表現する要素
- (3) ページ間の関係を表現する要素

以降、各要素についての詳細を述べる。

(1) 主体となる情報

主体となる情報は、ページにおいて最も価値のある情報であり、かつ閲覧者が最終的に求める情報である。要素の構成方法は様々であるが、本論文では構成に関係なく、主体となる情報の全てを「コンテンツ」と呼ぶ。

(2) 列挙を表現する要素

ある複数かつ同類の要素が集合し、それぞれの要素内に他のページへのリンクを含んだ上、一定の構成で並べられることがある。本論文では、これを「リスト」と呼ぶ。図 4.1 に、リストの HTML レンダリングの例を示す。



図 4.1: リストの HTML レンダリングの例

図 4.1 の点線で囲まれた要素は、複数のニュースの見出しを全て同じ形式で列挙している。また、それぞれの見出しはニュースの詳細ページへのリンクを含んでいる。一方、図 4.2 のように、連続性を持つが、各要素内に他のページへのリンクを含まないものについてはリストと見なさない。

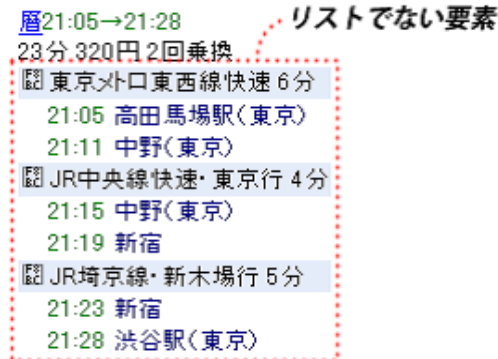


図 4.2: リストでない要素の HTML レンダリングの例

(3) ページ間の関係を表現する要素

コンテンツやリストは論理的な要素であり、物理的なページの構成とは無関係である。したがって、複数の物理的なページにまたがって1つの論理的なコンテンツやリストが存在しうる。そのような場合、コンテンツやリストが一定の量ごとに物理的なページに分割され、それらのページを互いに繋ぐ役割を持つリンクが存在することになる。本論文では、これを「ページング」と呼ぶ。図 4.3 に、ページングの HTML レンダリングの例を示す。

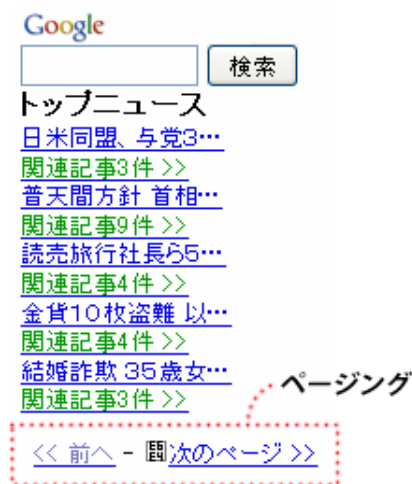


図 4.3: ページングの HTML レンダリングの例

図 4.3 では、点線で囲まれた部分の「前へ」「次のページへ」というリンクによって、その上部に配置されているリスト(図 4.1 に示した要素)が複数の物理的なページに渡って分割されていることを示している。このとき、ページングとリストが意味的に関連付いている。

以上が構成要素の分類であるが、分類上、「コンテンツ」はリスト、ページング以外の要素を示すことになる。図 4.4 に、コンテンツの HTML レンダリングの例を示す。図 4.4 では、点線で囲まれた部分がリスト(図 4.1 に示した部分)でもページング(図 4.3 に示した部分)でもないため、コンテンツとなる。



図 4.4: コンテンツの HTML レンダリングの例

リストの制限

列挙を表現する要素のうち、意味的にコンテンツと見なされるものがある。例えば、図 4.5 において、点線で囲まれた部分にはリンクを含む同類の要素が並べられており、リストであると言えるが、その内容は Web サイトのメニューを示す情報であることから、これはコンテンツとも言える。このような曖昧な要素を区別するために、リストの分類として、ページングと意味的に関連付いた列挙要素のみを「リスト」とする。これにより、「リスト」はある項目の索引(項目の詳細ページへのリンク)として表現される要素に限定される。また、リストは必ず複数のページに渡って分割されていることになる。



図 4.5: コンテンツと見なされる要素の HTML レンダリングの例

4.2 Web 文書構造の定義

実際に Web ページから構成要素を抽出するためには、要素の具体的な Web 文書構造を把握する必要がある。そこで、4.1 節で分類した各構成要素の Web 文書構造について、以下のように定義した。

ページングの Web 文書構造

ページングは複数ページ間の関係を表現する要素であり、特定の文書構造を持つわけではない。したがって、次の全ての条件を満たす文書構造が存在する場合、これをページングと判断する。

- (1) リンク (アンカータグ) として存在する。
- (2) 同一ページ内に、意味的に関連付いたコンテンツまたはリストが存在する。
- (3) リンク先の URL が、現在のページの URL と部分的なパラメータのみ異なる。

リストの Web 文書構造

Web ページ内の特定の文書構造が同一ページに連続して存在し、その各構造内に他のページへのリンクを含む場合、この構造群をリストと判断する。また、リ

ストの存在するページには、このリストと意味的に関連付いたページングが存在する。図 4.6 に、リストの文書構造の例を挙げる。

```
<div>  
  <div>  
      
    <a href="item1.html">Item1</a>  
  </div>  
  <div>  
      
    <a href="item2.html">Item2</a>  
  </div>  
  <div>  
      
    <a href="item3.html">Item3</a>  
  </div>  
</div>
```

図 4.6: リストの Web 文書構造の例

図 4.6 では、最上位要素の子の DIV 要素 (点線で囲まれた部分) がリストの項目の一つであり、これと同一形式の要素が連続した構成になっている。

コンテンツの Web 文書構造

リスト、ページング以外の要素がコンテンツとなるため、リストでもページングでもない全ての文書構造をコンテンツとする。コンテンツは、テキストや画像、他のページへのリンクなどを含む。また、連続性を持つがリストと見なされない文書構造もコンテンツとなる。

4.3 Web 文書構造モデルの定義

4.2 節で定義した Web 文書構造に基づき、Web 文書構造モデルのデータ構造を定義する。図 4.7 に、UML[7] のクラス図を用いて定義したデータ構造を示す。

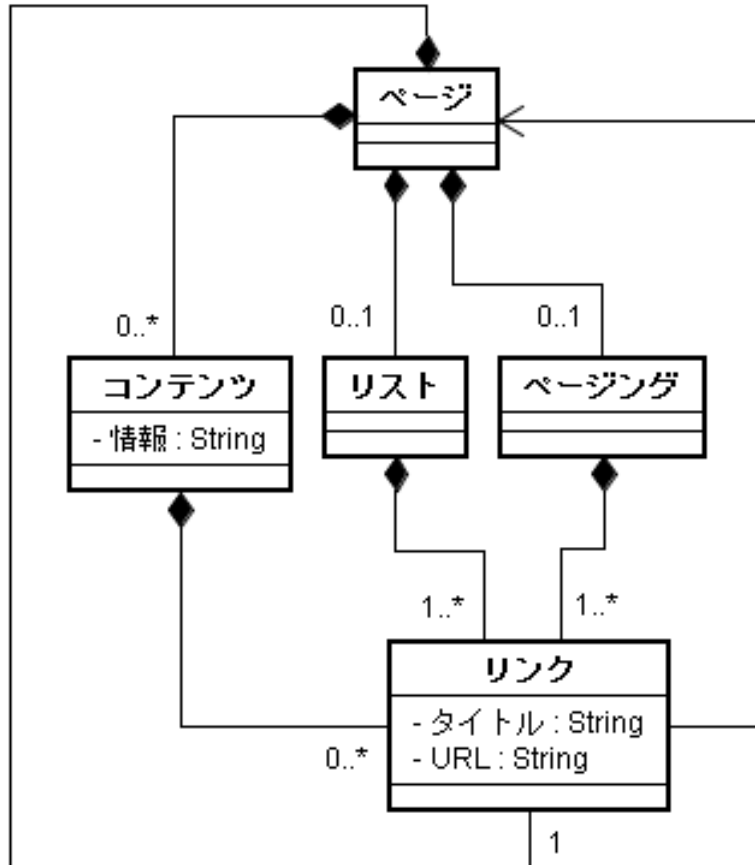


図 4.7: Web 文書構造モデルのデータ構造

ページは自身のページ情報 (タイトルと URL)、コンテンツとリスト、ページングを持つ。コンテンツは情報を持ち、その情報にはリンクが含まれている。一方、リストとページングはリンクの集合である。リンクはタイトルと URL を持った上で、特定のページに関連付いている。

第5章 通常端末向け Web ページの構成

本章では、携帯端末向け Web ページから文書構造を抽出し、これをモデルに変換する手順の詳細を述べる。さらに、生成されたモデルから通常端末向け Web ページを構成する手法について述べる。

5.1 前提条件

携帯端末向け Web ページから文書構造の抽出を行うにあたり、入力となる携帯端末向け Web サイトについて、次に挙げる項目を前提条件とする。

- (1) Web サイト全体で見ると、ページ間のリンクの繋がりが環構造であるが、トップページ(根)から末端ページ(葉)までの構造において木構造が抽出できると仮定する。図 5.1 に、この模式図を示す。図 5.1 において、環構造を点線、木構造を実線で示す。

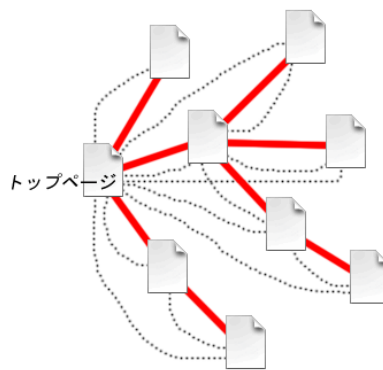


図 5.1: Web サイトにおけるページ間構造

- (2) 木構造の一部において、1つのコンテンツまたはリストがページ単位で分割され、ページングされた構成であると仮定する。図 5.2 の模式図において、点線で囲まれた部分は1つのコンテンツまたはリストであり、これが物理的にページ分割されていることを示す。

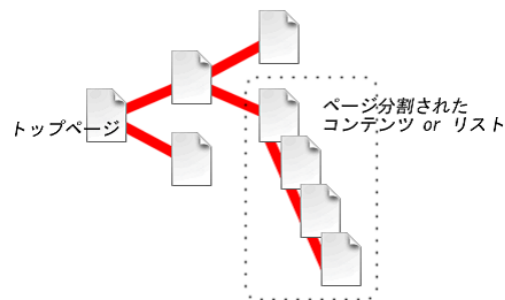


図 5.2: ページングされた構成

以上の条件は、本研究において調査した Web サイトを含め、既存のほとんどの携帯端末向け Web サイトが満たすものである。

5.2 Web 文書構造抽出とモデルへの変換

携帯端末向け Web ページから文書構造を抽出し、モデルへ変換する手法について述べる。まず、変換における前提事項を以下に示す。

- (1) Web ページの文書構造について、マークアップの間違いなどの属人的な問題は変換に影響しないものとする。なお、我々の作業においては、対象となる Web ページに対して Tidy[8] を適用して解決している。
- (2) 「1つの Web サイト」の括りは目視で判断する。また、モデルへの変換は Web サイトのトップページから行うが、トップページの判断も目視で行う。
- (3) Web ページの移動は、現在参照しているページ内のリンクに従う。このとき、一度参照したページへは移動しない。

次に、変換の手順を以下に示す。また、手順の各項目において、4.3 節の定義を元に、生成されるモデルの例を UML のオブジェクト図を用いて示す。

- (1) Web サイト内のある 1 ページを選択し、このページに対応するモデルを生成する。ここで、例として選択したページの HTML レンダリングを図 5.3、このページに対応するモデルの例を図 5.4 に示す。

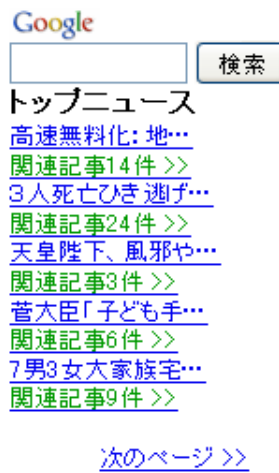


図 5.3: 選択したページ

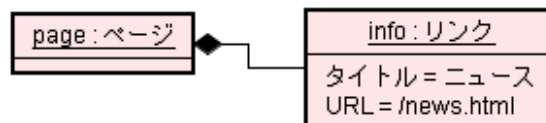


図 5.4: 選択したページのモデル

- (2) ページの文書構造を走査し、ページングの有無を調べる。ページングが存在する場合、これに含まれる全てのリンクを記憶しておく。この状態におけるページの HTML レンダリングを図 5.5、このページに対応するモデルの例を図 5.6 に示す。

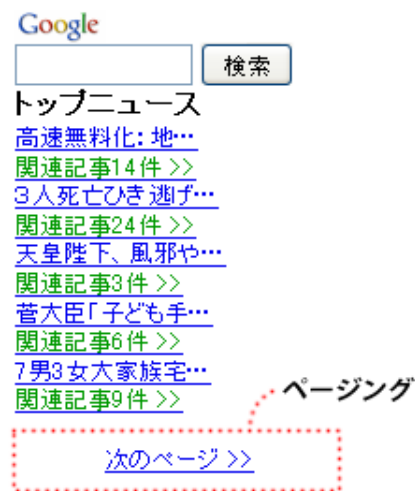


図 5.5: ページングが存在するページ

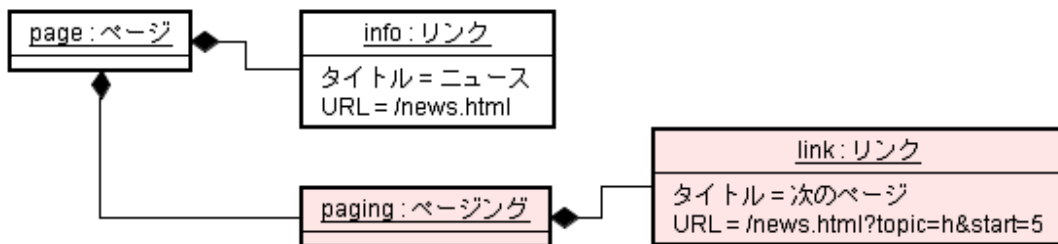


図 5.6: ページングが存在する場合のページのモデル

- (3) ページングがあった場合、これと関連付いたコンテンツまたはリストの有無を調べる。コンテンツまたはリストが存在する場合、これに対応するモデルを生成する。図 5.7 では、リストが存在するページの HTML レンダリングを示す。また、このページに対応するモデルの例を図 5.8 に示す。

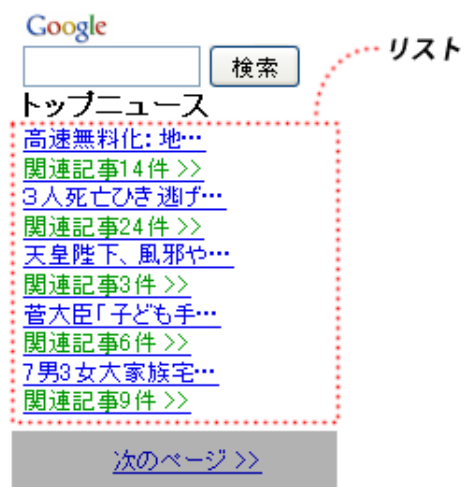


図 5.7: リストが存在するページ

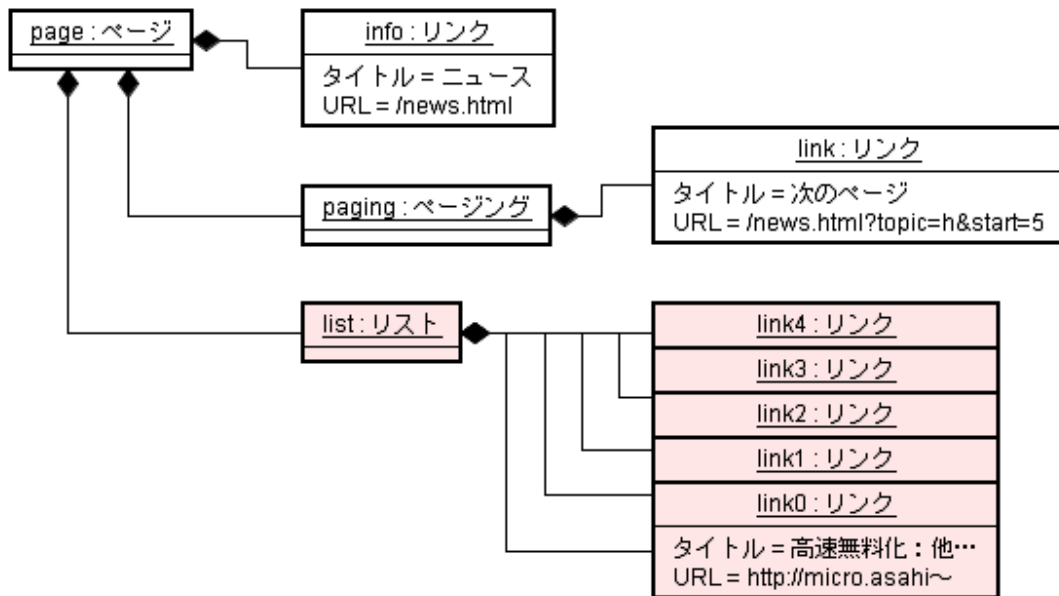


図 5.8: リストが存在する場合のページのモデル

- (4) ページ内の残りの文書構造を、既にモデルが生成された要素を境界として1つ以上のコンテンツと見なす。各コンテンツに対し、対応するモデルを生成する。この状態におけるページの HTML レンダリングを図 5.9、このページに対応するモデルの例を図 5.10 に示す。



図 5.9: コンテンツが存在するページ

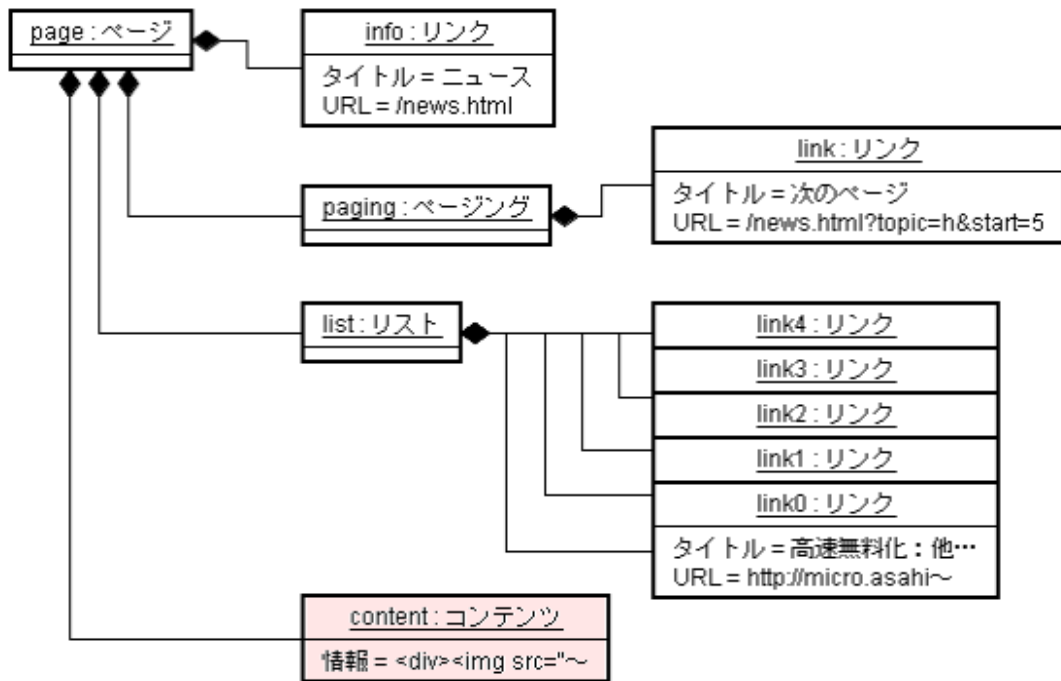


図 5.10: コンテンツが存在する場合のページのモデル

- (5) ページを移動する。移動先はページング内のリンクが最も優先度が高く、以降、コンテンツ内のリンク、リスト内のリンクの順とする。ページ移動後の手順は(1)に戻る。

また、ページング内のリンクから移動した先のページ(図 5.11)を参照する場合、ページングされたコンテンツまたはリストのモデルをリンク元ページのモデルに結合するため、前述の手順(n)に対し、以下の手順(n')を追加する。

- (2') リンク元ページのモデルにおいて、記憶するページングのリンクを更新する。この状態におけるモデルの例を図 5.12 に示す。
- (3') リンク元ページのモデルに、生成したコンテンツまたはリストのモデルを結合する。図 5.13 に、リストを結合したモデルの例を示す。
- (4') リンク元ページに存在するコンテンツと同一の要素については、対応するモデルを生成しない。



図 5.11: ページング内のリンクから移動した先のページ

以上の手順を繰り返すことにより、Web サイトの各ページに対応するモデルが生成される。Web サイト内の全ページのモデルが生成されたとき、これが Web サイトに対応するモデルとなる。

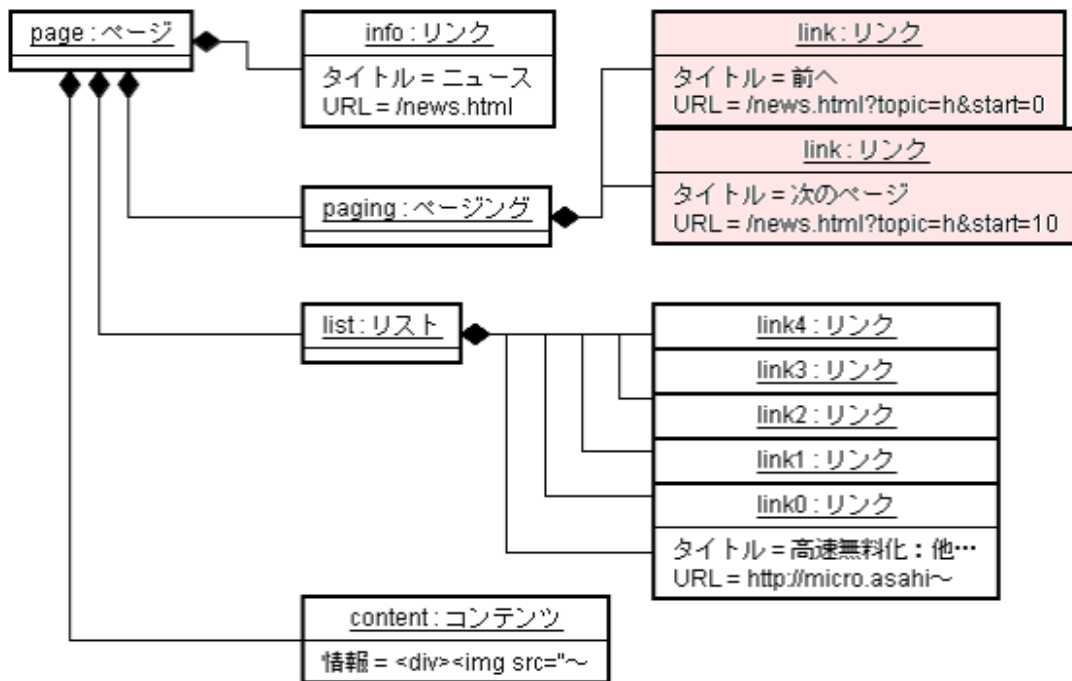


図 5.12: ページングのリンクを更新した場合のページのモデル

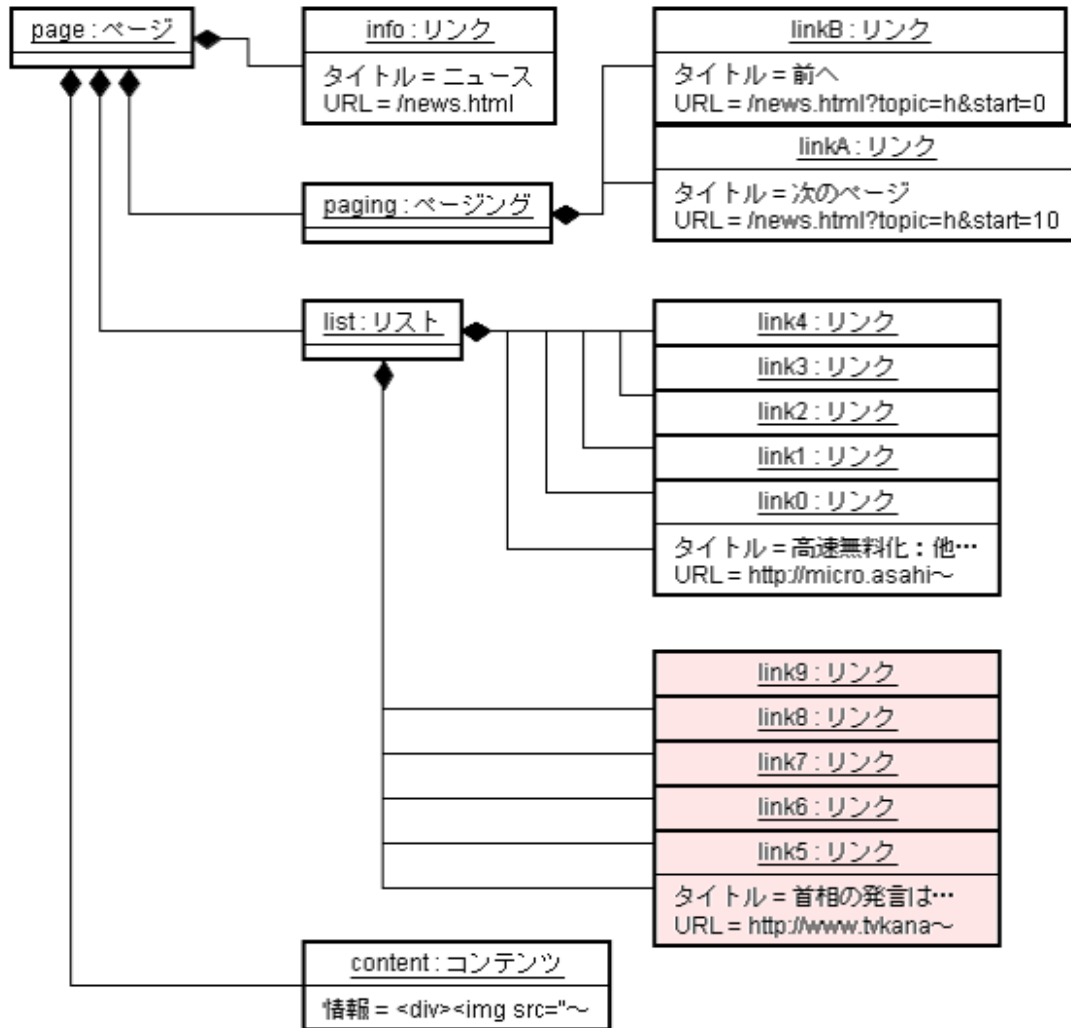


図 5.13: リストを結合した場合のページのモデル

5.3 通常端末向け Web ページの構成

生成されたモデルにはデバイス非依存の情報のみが含まれているため、モデルを利用してデバイス能力に応じた Web ページを構成することができる。

モデルを手法の外部で利用する方法

手法内部におけるモデルのデータ構造を外部に出力するためのスキーマ定義を付録 A に示す。

本手法に付属の Web ページ構成手法

モデルを Web ページへ変換するために、モデルを展開するための情報が必要になる。そこで、モデル以外の入力として、本手法に以下のレイアウト情報を指定する。

- (1) 出力対象のデバイス (例：通常端末)
- (2) リストの構成方法 (例：UL タグと LI タグで構成)
- (3) コンテンツの構成方法 (例：DIV タグで構成)

以上の入力により、本手法がページ上にモデルを展開し、リンク制御の付加を行う。この処理は指定した構成方法と出力対象デバイスの能力に応じて行う。デバイスが通常端末の場合、1 つの Web ページに全てのモデルを展開し、CSS によって要素の非表示とする。さらに、アンカーのクリックによってページ単位での表示切り替えがされるように JavaScript で制御する。この通常端末向け Web ページの例を図 5.14、図 5.15、図 5.16 に示す。図 5.14 では、初期状態として入力された携帯端末向け Web サイト [9] のトップページに対応する要素のみを表示している。アンカーのクリックによって図 5.15、図 5.16 の順に、遷移無しで情報が表示されるようになっている。

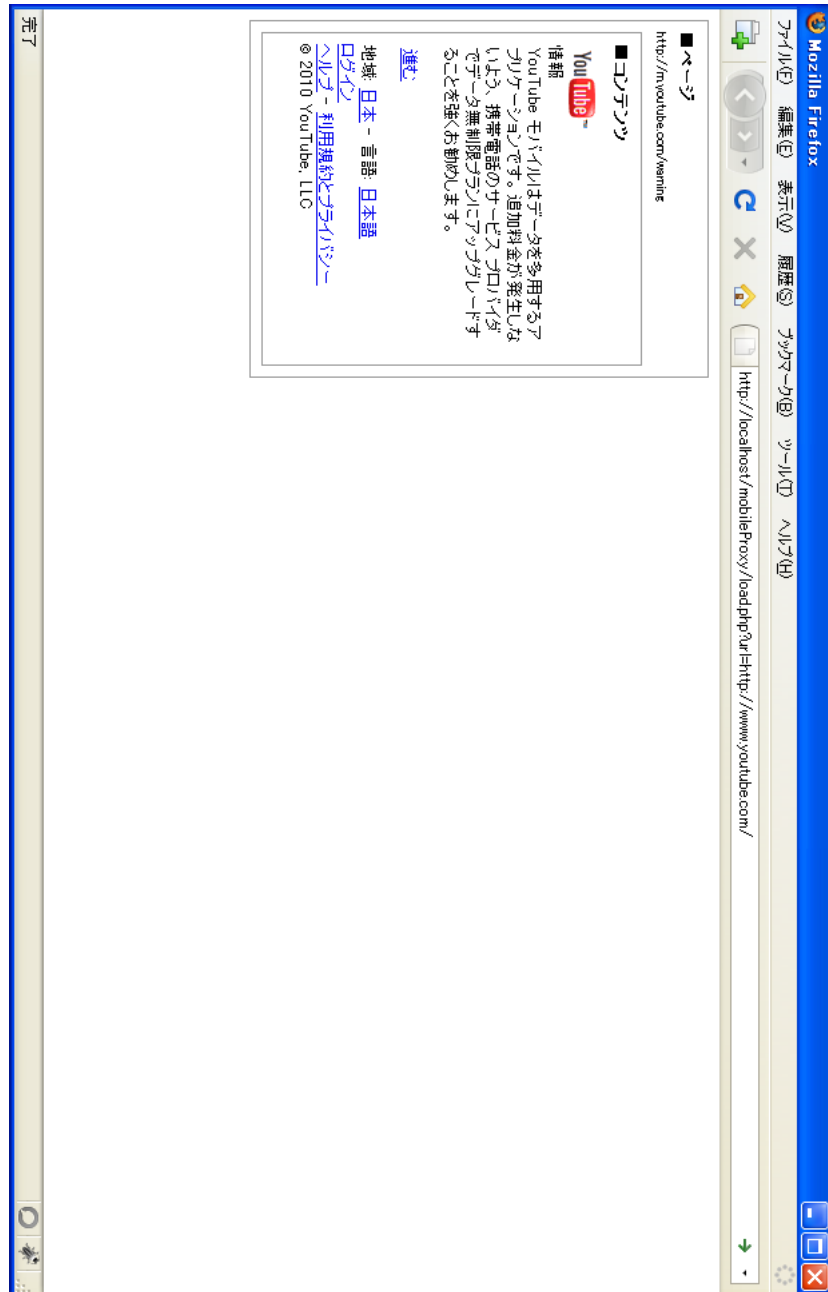


図 5.14: 通常端末向け Web ページの例 (1 ページ目の表示)

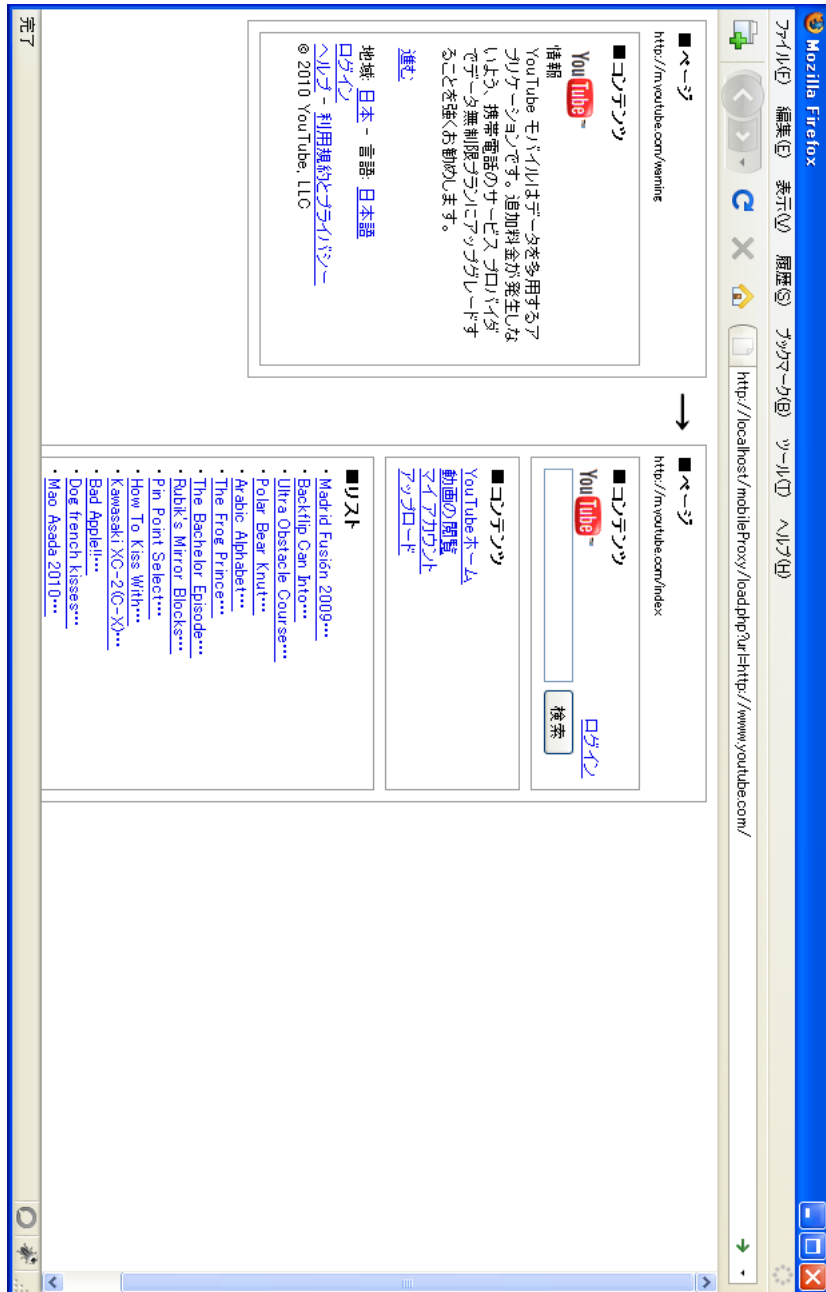


図 5.15: 通常端末向け Web ページの例 (2 ページ目の表示)



図 5.16: 通常端末向け Web ページの例 (3 ページ目の表示)

第6章 評価

本章では、本手法の評価とその結果について述べる。

6.1 評価手法

まず、本手法を用いて既存の携帯端末向け Web サイトから通常端末向け Web ページを生成する。次に、両者を通常端末から閲覧し、Web サイト内の全てのページ(要素)を辿るまでのページ遷移数を比較する。これにより、生成された通常端末向け Web ページの快適性が、元の携帯端末向け Web サイトに比べてどの程度向上しているかを評価する。また、コンテンツの構成が異なる複数の Web サイトに対して評価を行うことで、内容の違いによる快適性向上率の差も調査する。

6.2 結果

本手法の入力となる携帯端末向け Web サイトとして、一般に公開されている中から3つの Web サイト(ニュースサイト、動画投稿サイト、SNS サイト)を選択した。なお、各 Web サイトは4.1 節における調査では対象としなかったものである。どの Web サイトも目視で全容が確認できる規模であるが、ページングされているページ群については Web サイトの情報量の差が評価結果に影響しないようにするため、その先頭ページから最大で5 ページ目までを評価の対象とした。

また、携帯端末向け Web サイトを閲覧する際には、トップページから閲覧を開始した後、Web サイト内の全てのページを辿るまでのページ遷移数を計測した。一方、通常端末向け Web ページの生成は5.3 節に述べた例と同様の手法で行ったため、ページの遷移が発生しない。そこで、遷移数の代わりとして、トップページから閲覧を開始した後、全ての要素が表示されるまでのリンクのクリック数を計測した。両者において、一度通過したページへの移動はカウントしない。

以上の条件によって計測した結果を表 6.1 に示す。

表 6.1: 全てのページを辿るまでのページ遷移数の計測結果

| 入力 | 遷移数 (既存) | 遷移数 (生成) | 遷移減少率 |
|---------|----------|----------|--------|
| ニュースサイト | 76 | 55 | 約 28 % |
| 動画投稿サイト | 121 | 84 | 約 31 % |
| SNS サイト | 238 | 148 | 約 38 % |

表 6.1 には、既存の携帯端末向け Web サイトにおける遷移数、生成した通常端末向け Web ページにおける遷移数、前者に対する後者の遷移減少率 (小数第 3 位で四捨五入) を示している。結果として、全ての Web サイトで通常端末向け Web ページの遷移数が減少した。また、情報が多様な Web サイトの方が遷移数の減少幅が大きくなった。

6.3 考察

生成した通常端末向け Web ページの遷移数が減少したのは、全ての携帯端末向け Web サイトがページングによってページあたりのコンテンツやリストの量を制御しており、それが本手法によって解除されたためである。今回対象とした全ての Web サイトにおいてページングが用いられていた。その中で、多様な情報を含む Web サイトの方がページングの割合が多いために、遷移数のより大きな減少を生じたと考えられる。

また、今回の評価は生成後の Web ページの方が快適性が高いという前提で行った。ところが、ページングされたコンテンツやリストを全て連結することにより、生成した Web ページの構成が縦に伸びてしまうことが判明した。今回はページングされたページは最大で 5 ページのみを評価の対象としたために問題はなかったが、連結する要素が多くなると遷移数以外の指標を考慮する必要がある。

第7章 関連研究

本章では、本研究の大枠である「特定のデバイス向けに特化した Web ページを、デバイス能力に応じた Web ページに変換する手法」を提案する既存研究について述べる。以下に、既存研究を提案手法のアプローチ別に分けて述べ、本研究との比較を行う。

(1) ページ分割 (Page Segmentation) による再構成

PC 向けの Web ページを特定のアルゴリズムによって分割、またはコンテンツを抽出し、スマートフォンや携帯電話向け Web ページを再構成する。文献 [10] では、ページ内の DOM の位置情報を元に、決定木を用いてページを分割する手法を提案している。また、文献 [11] では、タグの数や深さから算出されるコンテンツ間の距離を元にした分割手法を提案している。本研究では逆に、携帯端末向けの Web ページの合成により、画面領域の大きな端末向け Web ページを生成する手法を提案している。

(2) デバイス非依存の Web ページ構築

開発者はデバイス共通の Web ページを構築し、サーバー上の Web システムがこれをクライアントのデバイス能力に応じたページに変換する。文献 [12] では、JSP のカスタムタグを利用した、デバイスの画面サイズに依存しない Web ページの構築手法を提案している。本研究では、Web 文書構造モデルがデバイス非依存であるが、これは手法内部における中間状態である。

(3) デバイスに適した Web コンテンツの選択

Web におけるデバイス独立 (Device Independence[13]) とは、Web コンテンツがデバイスの種類にかかわらず閲覧可能であることをいう。これを実現するために提案された仕様に、CC/PP[14] がある。これは、デバイス情報をサーバーとクラ

クライアントに持たせることで、サーバー上の Web システムがクライアントのデバイス能力に応じたコンテンツの出力を可能にするための枠組みである。文献 [15] では、CC/PP を利用したビュー (View) の制御フレームワークを提案している。

本研究では、デバイス情報を手法外部からの入力としており、本手法がデバイス能力の判別を行うわけではない。もし本手法を実装する場合、リクエストしてきたデバイスに応じて処理を選択するような機構を設けるのであれば、このような仕組みが必要になると考えられる。

第8章 おわりに

本論文では、携帯端末の小型画面向けに特化した Web ページから通常端末の画面向け Web ページを構成する手法を提案した。携帯端末向け Web ページをモデル化することで、デバイスの画面サイズに依存しない Web ページの生成が可能になった。また、既存の携帯端末向け Web サイトに対して評価を行い、閲覧における快適性の向上を確認した。以下、本研究の今後の課題について述べる。

文書構造抽出アルゴリズムの改善

本論文で定義したページングとリスト、コンテンツには構造的な関連付けがない。したがって、目視で意味的に関連を判別することはできても、構造から関係を読み取ることができない。文書構造の抽出処理をプログラムによって実装する場合には、これを解決する手法を考案しなければならない。

モデルの定義拡張

現在のモデルの定義では、リストとページング以外の要素を全てコンテンツとしているが、これは主体となる情報としては非常に曖昧である。また、情報量が通常端末向け Web ページに比べて相対的に少ない構成であることから、モデル化の際にページ概念を除去することで、情報を集約、再構成することも考えられる。以上の点で、情報という観点からモデルの定義を発展させる必要がある。

謝辞

研究を進めるにあたり、学部から3年間にわたってご指導、ご助言を頂いた深澤良彰教授、日本IBM 東京基礎研究所の小野康一氏に深く感謝致します。本研究には十分な時間を費やすことができませんでしたが、おかげさまで何とか形にはなりました。

また、苦楽を共にした深澤研究室の皆様、同期の皆様に深く感謝致します。研究活動において自分の支えになりました。

参考文献

- [1] Google
<http://www.google.co.jp/>
- [2] NTT ドコモ : iモードブラウザ仕様
<http://www.nttdocomo.co.jp/service/imode/make/content/browser/>
- [3] KDDI au : XHTML
<http://www.au.kddi.com/ezfactory/xhtmll.html>
- [4] SoftBank : WEB & NETWORK について
<http://creation.mb.softbank.jp/web/index.html>
- [5] W3C : HTML 4.01
<http://www.w3.org/TR/html4/>
- [6] W3C : XHTML 1.0
<http://www.w3.org/TR/xhtml1/>
- [7] OMG : UML
<http://www.uml.org/>
- [8] W3C : Tidy
<http://www.w3.org/People/Raggett/tidy/>
- [9] YouTube
<http://www.youtube.com/>
- [10] Shumeet Baluja : "Browsing on Small Screens: Recasting Web-Page Segmentation into an Efficient Machine Learning Framework", WWW 2006.
- [11] Gen Hattori, Keiichiro Hoashi, Kazunori Matsumoto, Fumiaki Sugaya : "Robust Web Page Segmentation for Mobile Terminal Using Content-Distances and Page Layout Information", WWW 2007.

- [12] Giani Carla Ito, Mauricio G. Ferreira, Nilson Sant'Anna : "A Strategy of Development for Web Interfaces in Mobile Devices", IEEE Latin America Transactions 2008.
- [13] W3C : Device Independence
<http://www.w3.org/2001/di/>
- [14] W3C : CC/PP
<http://www.w3.org/Mobile/CCPP/>
- [15] Jochen Muller, Torsten Lenhart, Dirk Henrici, Markus Hillenbrand, Paul Muuller : "A Developing Web Applications for Mobile Devices", DFMA '05.
- [16] W3C : XML Schema
<http://www.w3.org/XML/Schema>

研究業績

勝俣浩樹, 小野康一, 深澤良彰, 携帯端末用 Web ページから通常端末用ページを構成する手法, 電子情報通信学会ソフトウェアサイエンス研究会, 2009 年 12 月研究会, 2009.

付録A Web文書構造モデルのスキーマ定義

本章では、4.3節で定義したWeb文書構造モデルのスキーマ定義を示す。スキーマ定義にはXML Schema[16]を用いる。

```
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="page" type="Page" />
  <xsd:complexType type="Page">
    <xsd:sequence>
      <xsd:element name="link" type="Link" />
      <xsd:element name="contents" type="Contents"
        minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="lists" type="List"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType type="Contents">
    <xsd:sequence>
      <xsd:element name="content" type="Content"
        minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType type="Content">
    <xsd:sequence>
      <xsd:element name="data" type="xsd:string" />
      <xsd:element name="links" type="Links" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType type="List">
    <xsd:sequence>
      <xsd:element name="links" type="Links"
```

```
        minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType type="Links">
    <xsd:sequence>
        <xsd:element name="link" type="Link"
            minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType type="Link">
    <xsd:sequence>
        <xsd:element name="name" type="xsd:string" />
        <xsd:element name="value" type="xsd:string" />
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```