

Web 適応のためのユーザプロファイルの自動生成方式

神保 昌徳† 原 英樹† 菅原 研次†

†千葉工業大学大学院 情報科学研究科 情報科学専攻
‡千葉工業大学 情報科学部 情報ネットワーク学科

概要

本論文では、閲覧者のユーザプロファイルを自動生成し、それを利用して Web ページの変更を行う Web 適応システムの提案を行う。Web 作成者は Web サイト作成時に予想される閲覧者の種別とその判別方法を定義する。Web ページの閲覧時に閲覧者がどの種別に属するかを判断してユーザプロファイルを生成し、そのユーザプロファイルに基づいて Web ページを変更する。Web 適応に必要なルールは、Web ページ毎に記述せず Web サイトあるいは関連のある Web ページ群に対して記述する。そのため記述量の低減が期待できる。

A method of generating a user profile automatically for the Adaptive Web

Masanori Jimbo† Hideki Hara‡ Kenji Sugawara‡

Abstract

In this paper, we propose a web adaptation system which generates a user profile automatically and adapt web pages to the user based on the profile. A web designer predicts type of user who browse the pages and describes a template of user profile and rules for generate the profile. When a user browse the web pages, our system generates the profile and adapt the web pages based on the profile. The rules for web adaptation is not described for a web page but a group of web pages. So, size of rules for web adaptation will be reduced.

1 はじめに

近年、ECをはじめとするインターネットを利用した多様なアプリケーションが普及し、膨大な量の Web ページが公開されている。一方、Web ページの閲覧者の増大に伴い、仕事や日常生活などの利用状況も多様化してきており、個々の閲覧者が必要としている情報や利用環境などに合わせた Web データの変更処理を行う、いわゆる Web 適応の研究が盛んに行われている [1, 2, 5, 6, 7, 8, 9]。

Web 適応の研究の一つに、閲覧者に提示する Web ページの内容を動的に変化させる機構が提案されている [3]。これは、Web ページを変化させるために必要なルール記述を Web ページ毎に行うことで実

現されている。しかし一つの Web サイトは通常、複数の Web ページで情報提供を行っている。そのため、提案されている方式で変化させることを考えた場合、Web ページ毎にルール記述を行うことが必要となり、Web 作成者の負担となる。

本論文では、Web 適応に必要なルールの記述量を低減することを目的としたシステムを提案する。予想される閲覧者の種別をあらかじめ定義しておき、閲覧時に閲覧者がどの種別に属するかを判断してユーザプロファイルを作成し、そのユーザプロファイルに基づいて変更した Web ページの提示を行う。Web 適応のために必要なルール群を Web サイトあるいは関連のある Web ページ群に対して記述するため、記述量の低減が期待できる。

2 Web 適応と関連研究

2.1 Web 適応

Web データを利用する状況や環境に合わせて変化させることを Web 適応と呼ぶ [2].

例えば、閲覧者の興味によって Web ページの内容を変化させることが考えられる。ショッピングサイトでは閲覧者が興味を持つと考えられる商品に関しては詳細まで述べ、異なる商品に関しては概略程度に留めた表記に変化させる。よって、閲覧者は各分野の知識に合わせて購入することができる。

しかしながら Web ページは通常、閲覧者がどのように閲覧しても変化しない静的な情報である。Web ページを利用する状況や環境に合わせて変化させる方式の一つに、Web ページ毎に CGI などの手続き型言語を用いて構築する方式がある。しかし、各 Web ページ毎に手続き型言語による構築を行うことは効率的であるとは言えない。そこで、Web ページまたは Web サイト毎に宣言的な言語を記述することで、Web ページを利用する状況や環境に合わせて変更する機構があれば有用であると考えられる。

2.2 関連研究

本研究と関連する主な研究として、ActiveWeb がある [3]。ActiveWeb は利用している日付や場所などの状況に合わせて Web コンテンツを変化させるシステムである。Web 作成者は適応する状況を想定し、ECA ルールと呼ばれるルールを模した宣言的な言語を Web コンテンツに対して記述する。しかしながら、我々は ActiveWeb に以下の問題点があると考えた。

- Web サイトの規模に伴うルールの記述量の増加
ActiveWeb が提案するルールは Web コンテンツ毎に記述する。しかし一つの Web サイトは通常、複数の Web ページで情報提供を行っている。そのため、同様の処理を複数の Web ページで行う際も Web コンテンツ毎に記述しなければならない。
- Web 適応のための変更能力が不十分
ActiveWeb ではルール内で Web ページ中の変更対象を指定するが、一つのルールでは指定する対象は一つのみである。そのため、複数の対象を操作する場合はルールを複数記述しなければならない。

以上により、現実的な Web サイトに ActiveWeb を適用する場合、記述すべきルールは膨大となり、

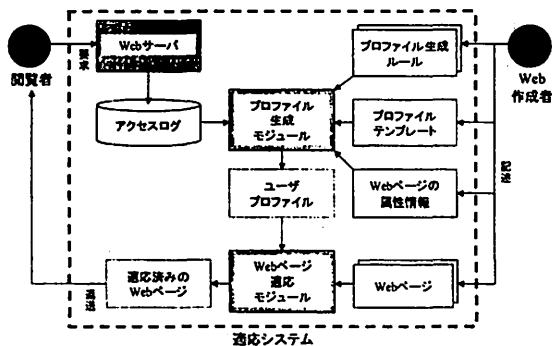


図 1: 本システムの概要

Web 作成者の負担は増加してしまう。そこで、本研究ではユーザプロフィールの自動生成とそれを利用した Web 適応の実現方式について提案する。自動生成されたユーザプロフィールを Web サイト内の全ての Web ページで利用することで、Web 作成者の Web 適応に関する手続きの記述量を減らし、負担を軽減することができる。

3 ユーザプロフィールを用いた Web 適応

3.1 適応手法の概要

本論文の適応手法の概要を図 1 に示す。

Web 作成者は Web サイト構築時に、プロフィール生成ルール、プロフィールテンプレート、Web ページを作成する。プロフィール生成ルールは、閲覧者のユーザプロフィールの生成方法を記述したものである。プロフィールテンプレートは、生成されるユーザプロフィールの雛型である。Web ページは、適応の対象となる HTML ソースである。

Web 適応を行うために本システムは二段階の動作を行う。最初の段階ではプロフィール生成モジュールがアクセスした閲覧者のユーザプロフィールの生成を行う。閲覧者がアクセスすると、その内容はアクセスログに蓄積される。用意されたプロフィールテンプレートを元に、アクセスログの情報とプロフィール生成ルールを比較し、ユーザプロフィールの生成を行う。また、ユーザプロフィールの生成制度を高めるために、Web ページの属性情報と呼ばれる情報も併せて用いる。

二段階目では前段階で生成されたユーザプロフィールを用いて、Web ページの変更モジュールが Web ページの変更を行う。変更され、閲覧者に適応済みとなった Web ページは閲覧者へと返送される。

以下では、各段階で用いられるドキュメントやタ

```
terminal = pc
literacy = beginner
```

図 2: ユーザプロフィールの例

```
terminal = { pc, cell-phone }
literacy = { expert, beginner, other }
```

図 3: プロファイルテンプレートの例

グについて説明を行う。

3.2 ユーザプロフィールとモデリング

3.2.1 ユーザプロフィールと プロファイルテンプレート

本システムでは、Web 適応のために閲覧者像の推測を行う。推測された閲覧者像をユーザプロフィールと呼ぶ。ユーザプロフィールは閲覧者の属性とその属性値を一組とした項目の集合から成る。ユーザプロフィールの例を図 2 に示す。図 2 の場合、推測された閲覧者像は属性 terminal と属性 literacy の 2 項目から成る。

ユーザプロフィールは用意された雛型を元に生成される。この雛型をプロファイルテンプレートと呼ぶ。プロファイルテンプレートは閲覧者の属性とその属性値の候補値を一組とした項目の集合から成る。プロファイルテンプレートは Web 作成者が XML ドキュメントを用いて記述する。プロファイルテンプレートの例を図 3 に示す。図 3 の場合、プロファイルテンプレートは属性 terminal と属性 literacy の 2 項目から成っている。

プロファイルテンプレートに記述された属性の各候補値のうち、一つの属性値に定めたものがユーザプロフィールとなる。よって、図 3 のプロファイルテンプレートを雛型として生成されるユーザプロフィールの一例が図 2 となる。

3.2.2 プロファイル生成ルール

ユーザプロフィールの生成方法を宣言的な言語を用いて記述したものをプロファイル生成ルールと呼ぶ。プロファイル生成ルールは以下の 2 項目を一組としたルールから成る。

- 条件部

閲覧者のアクセス、アクセス履歴から導出される情報に対する条件を記述する。

```
<rule>
<conditions>
  <condition type="now" object="browser"
    op="like" value="DoCoMo"/>
</conditions>
<actions>
  <action order="set" object="terminal"
    value="cell-phone"/>
</actions>
</rule>
```

図 4: プロファイル生成ルールの記述例

- アクション部

条件を満たしたときに実行するユーザプロフィールを生成するためのアクションを記述する。

プロファイル生成ルールは XML ドキュメントを用いて記述する。プロファイル生成ルールの記述例を図 4 に示す。

図 4 のプロファイル生成ルールは、「現在利用している Web ブラウザに DoCoMo という文字列が含まれている場合、閲覧者の属性 terminal を cell-phone に設定する」というユーザプロフィールの生成方法を記述したものである。よって、DoCoMo の携帯電話を用いてアクセスした時、閲覧者の属性 terminal が cell-phone であるユーザプロフィールが生成される。

3.2.3 Web ページの属性情報

ユーザプロフィールを生成するために用いる Web ページの属性について記述したものを Web ページの属性情報と呼ぶ。

Web ページの属性情報は Web ページの相対パス、Web ページの属性、その属性値を一組とした項目から成る。また Web ページの属性や属性値は Web 作成者が任意に記述することができる。Web ページの属性情報は W3C が提唱する RDF[4] の形式に準じたドキュメントで記述する。

3.3 Web ページの適応

本システムは閲覧時にユーザプロフィールを生成した後、それに応じた Web ページの変更を行う。Web ページを変更させるためには、図 5 のようにその Web ページに新たなタグを付加する必要がある。

本システムでの Web 適応の手法は、修正や変更を Web ページに加えるのではなく、あらかじめ用意

```

01: <html><body>
02:   Taro's Macintosh<hr/>
03:   LC630<br/>
04:   <Adaptive item="literacy">
05:     <candidate value="beginner">
06:       For Office.
07:     </candidate>
08:     <default>
09:       CPU:33MHz, Memory:12MB
10:     </default>
11:   </Adaptive>
12: </body></html>

```

図 5: Web ページの記述例

されている選択肢の中から生成したユーザプロフィールに応じた部分を選択する方式を用いる。選択肢と成るのは `<Adaptive>`, `<candidate>`, `<default>`, これら 3 つのタグで囲まれた HTML ソースである。図 5 の場合、網掛けされた 2 つの HTML ソースを指す。Web 適応は以下の工程を繰り返すことで行う。

1. Adaptive タグの検出

記述順に Adaptive タグの検出を行う。図 5 の場合、4~11 行目を属性 `item` の属性値が `literacy` である Adaptive タグが囲んでいる。これは 4~11 行目の HTML ソースが閲覧者の属性 `literacy` の属性値を参照して適応することを示す。

2. 適切な HTML ソースの選択

Adaptive タグが囲む中から、適切な HTML ソースを選択する。適切な HTML ソースとは、前工程で参照した閲覧者の属性値と等しい値の属性 `value` を持つ `candidate` タグが囲む HTML ソースを指す。前工程で参照した閲覧者の属性は `literacy` であるので、図 2 の場合、その属性値は `beginner` である。属性 `value` が `beginner` であるのは 5~7 行目の `candidate` タグであるので、それが囲む HTML ソースが選択される。適切な HTML ソースが見つからない場合、`default` タグが囲む HTML ソースが選択される。

3. Web ページの変更

選択された HTML ソース以外の HTML ソースを隠蔽する。また、付加した各タグも隠蔽される。よってユーザプロフィールが図 2 である閲覧者の場合、図 6 の Web ページが提示される。

```

<html><body>
  Taro's Macintosh<hr/>
  LC630<br/>
  For Office.
</body></html>

```

図 6: 適応済みの Web ページの例

4 実装と評価

提案した方式に従い、プロトタイプシステムの実装を行った。実装したシステムは WindowsXP を OS として使用し、JakartaTomcat に内蔵されている Web サーバ機能を利用している。提案した各モジュールは J2SE5.0 をコンパイラとした Java クラスで構築した。閲覧者 ID はクエリ情報に格納する。また、現行の Web ブラウザでは場所を送信する機能が実装されていないため、これも仮想的にクエリ情報に地域名を格納するものとする。結果、実験した本論文中で例示した Web 適応の動作が実現できることを確認した。

以下で、適応 Web として動作する Web サイトを構築する際に、提案方式と ActiveWeb の方式での作業負担の比較を行う。1 つの Web サイトにつき、 N 個の Web ページがあり、この全てが適応動作により変化するものとする。

本方式では最初に、プロフィールテンプレートを記述する。テンプレートの属性の項目数を n 、テンプレートの属性値の平均個数を T 個とすると、このプロフィールテンプレートからは、 nT 種類のプロフィールが生成される。本方式では、プロフィールの種類の数だけ Web ページが変化する。よって、Web ページの平均バリエーション Var は以下の式で表すことができる。

$$Var = nT$$

このバリエーションを実現するために必要な作業量を、適応条件の記述量と、Web ページの変更手続きの記述量とでそれぞれ比較を行う。

(a) 条件の記述量の比較

Web 適応の動作のためには、アクセスしている閲覧者をルールなどの条件記述により分類するプロセスが必要となる。この条件記述の記述量について比較を行う。

提案システムの場合の条件記述は、プロフィールテンプレート内のある 1 つの属性の属性値候補群から 1 つの属性値を選択するのに用いる。よって、1

つのプロファイル項目に対応するルール数の平均個数を r とすると、ルールの記述数も $nrT = rVar$ 個となる。

一方 ActiveWeb の方式は、条件記述はアクセスとアクセス履歴から成り、そのアクション記述に Web ページに対する操作内容を記述する。Web ページのバリエーションは個々のアクション記述の実行により実現される。したがって、あるバリエーションを実現するのに必要な平均実行回数を a とすると、Web ページの総数 N について Var 個のバリエーションを実現するためには、 $aNVar$ 個のルールが必要となる。

ActiveWeb の方式では、サイトを構成する Web ページの総数 N に比例してルールの記述量が增大するが、提案方式では Web ページの総数には左右されない。したがって現実的な Web サイトを作成する際に適応させる Web ページが増えた場合、本方式を用いることで負担軽減が期待できる。

(b) Web ページ変更手続きの記述量の比較

次に、Web ページの見え方を変化させるために必要なページ変更手続きの記述量について比較を行う。

提案方式では、Web ページにはプロファイルテンプレートの全てのバリエーションを記述する。ルールエンジンはこの中からテンプレートにあったものを選択し、表示する。バリエーションが増えるほど、記述する Web ページのサイズは大きくなり、記述コストは大きさに比例して増加する。バリエーション数を Var 、Web ページの総数を N とすると、労力は $NVar$ に比例する。

一方 ActiveWeb では、Web ページには、バリエーションを生じさせる記述は含まれない。よって、Web ページの個々の記述コストの平均を c 、Web ページの総数を N とすると、労力は cN になる。ただし ActiveWeb の方式では、ルールのアクションとして変更手続きを記述する。ルールの個数は (a) で示したように $aNVar$ となる。

二方式を比較した場合、バリエーション数とページ数に比例した記述量となることが推測される。

5 おわりに

Web データを利用する状況や環境に合わせて変化させるために、ユーザプロファイルの自動生成とそれを利用した Web 適応の実現方式を提案した。

ActiveWeb は全ての Web 適応に関する手続きを Web コンテンツ毎に記述するため、その記述量は膨大となる。対し本論文では、予想される閲覧者の種

別をあらかじめ定義しておき、閲覧時に閲覧者がどの種別に属するかを判断してユーザプロファイルを生成し、それに基づいて Web ページの変更を行う。予想される閲覧者の種別とその判断方法は Web サイトに対して記述する。よって、本研究は Web 適応に関する手続きの記述量を ActiveWeb と比較して低減させることが期待できる。

参考文献

- [1] Peter Brusilovsky, Mark T. Maybury: FROM ADAPTIVE HYPERMEDIA TO THE ADAPTIVE WEB, COMMUNICATIONS OF THE ACM, Vol.45, No.5, pp.31-33 (2002).
- [2] 清光英成: Web 適応:アクセスに基づく Web のダイナミックな変化, 情報処理, Vol.44, No.7, pp.689-695 (2003).
- [3] 清光英成, 竹内淳記, 田中克己: ActiveWeb:アクティブルールの XML 表現に基づく Web コンテンツの個別化とアクセス管理, 情報処理学会論文誌:データベース, Vol.42, No.SIG 8(TOD 10), pp.139-147 (2001).
- [4] W3C: Resource Description Framework (RDF), <http://www.w3.org/RDF/> (1999).
- [5] P. Atzeni, G. Mecca, P. Meriardo: *To weave the Web*, Proceedings of the 23rd International Conference on Very Large Data Bases, Athens, Greece (1997).
- [6] George Chang, Marcus J. Healey, James A. M. McHugh, Jason T. L. Wang, 武田善行, 梅村恭司, 藤井敦: Web マイニング, 共立出版, ISBN4-320-12087-6, pp.45-48 (2004).
- [7] 北山文彦, 広瀬紳一: Dharma:さまざまなインターネット端末にコンテンツを適応させるソフトウェア技術, 情報処理, Vol.42, No.6, pp.576-581 (2001).
- [8] IBM 東京基礎研究所: 多種端末向け Web アプリケーション構築技術:Dharma, <http://www.trl.ibm.com/projects/dharma/>
- [9] 灘本明代, 服部多栄子, 近藤宏行, 沢中郁夫, 田中克己: Web コンテンツの受動的視聴のための自動変換とスクリプト作成マークアップ言語, 情報処理学会論文誌:データベース, Vol42, No.SIG 1(TOD 8), pp.103-116 (2001).