

# 1

# Web適応

## ——アクセスに基づくWebのダイナミックな変化

清光 英成 神戸大学国際文化学部  
kiyomitu@kobe-u.ac.jp



Webデータを、利用する状況あるいは環境に合わせて変化させることをWeb適応 (Web Adaptation) という。

Mosaicが登場したころ、いわゆる初期のWebは、

- ページの内容 (コンテンツ) や体裁は誰がアクセスしても不変
- 設定されているリンクを巡行すれば無条件にリンク先ページへのアクセスが可能
- 過去にどのようなページを閲覧したか、あるいはどのようなリンク巡行をしたかにかかわらず、表示されるコンテンツとリンク巡行の結果は不変

のような静的な性質があった。

Web (あるいはインターネット) がさまざまな情報を共有できる情報社会基盤として、ユビキタスネットワーク環境の一翼を担うためには、Webデータを動的に変化させていつ・どこからでも最適な形式でコンテンツを送受信できることが求められる。Webデータの動的変化を実現する技術的な基盤を概観すると、CGI、SSI、JAVA ServletならびにMS-ASP (Microsoft Active Server Pages) などのサーバアプリケーション技術とともに、JAVA Appletに代表されるクライアントで実行されるアプリケーション技術やShockwave、Acrobatなどを実行するブラウザプラグインの技術によるところが大きい。このような技術の発達と浸透により、インタラクティブにWebデータを変化させることができるようになり、データベースから動的にページを構成するようなアプリケーションも可能となった。本稿では、ユーザや環境に応じてWebデータをダイナミックに変化させる適応化技術について解説する。

### アクセス機器や通信環境に基づくWebビューの適応

Webデータはデスクトップコンピュータから携帯情報端末まで、多様な機器からアクセスされ利用されている。デスクトップコンピュータと携帯電話のディスプレイでは、表示可能な大きさや解像度は明らかに異なる。また、ブロードバンド、LAN、電話回線やワイヤレス環境では、通信速度も大きく異なる。このような、端末の表示能力や通信速度などの物理的要因を考慮してWebデータの表現を変化させる要求は自然である。

一般的なWebページはデスクトップコンピュータでA4サイズ1枚程度の大きさになるように作られている。Webページは文字・数値データだけでなくカラー画像や映像などのマルチメディアデータをコンテンツとして含んでいる。このようなWebページにPDAなどの携帯端末でアクセスする場合、通信・コンテンツ処理・表示処理の能力の点で困難が多い。そこで、アクセス環境に基づいてコンテンツを適応化する場合、提供するWebデータの見せ方 (Webビュー) の再構成は以下の要因を判断材料にしている。

#### 通信環境

帯域：コンテンツの受信が満足にできるかどうか  
接続性：切断しやすい、切断しにくい

#### 端末の処理能力

CPUパワー：データ閲覧のためのパーズング処理  
に対して十分かどうか

表示画面：コンテンツを加工せずに表示可能かどうか

図-1に表示端末・通信回線を考慮した適応化の概念を示した。直観的には、端末・通信回線に応じて、デー

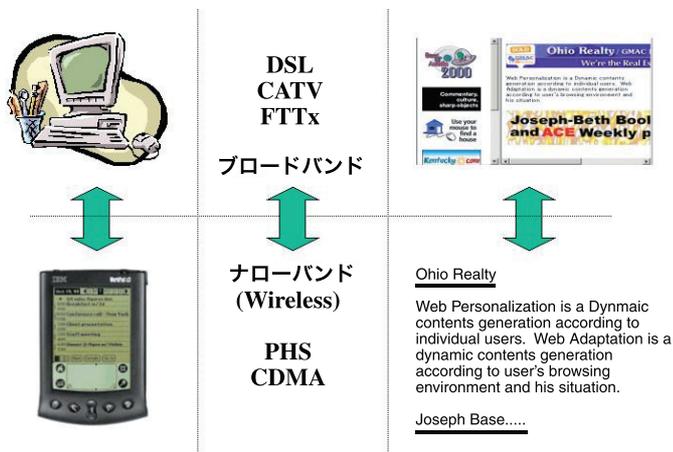


図-1 端末・通信環境に基づくコンテンツ適応化の例

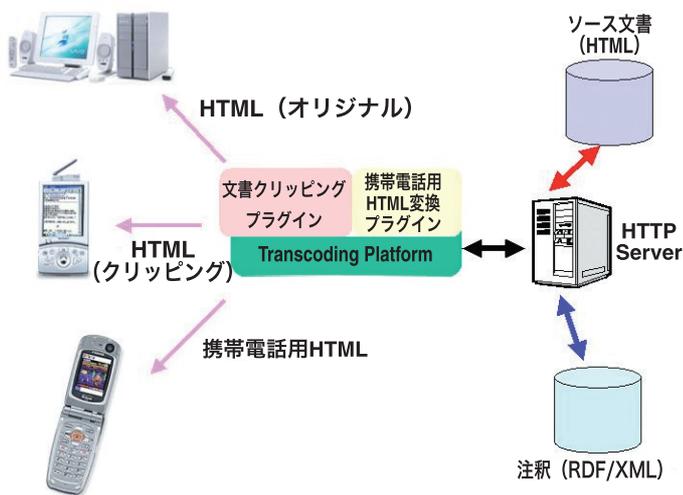


図-2 適応化実現のためのシステムアーキテクチャ

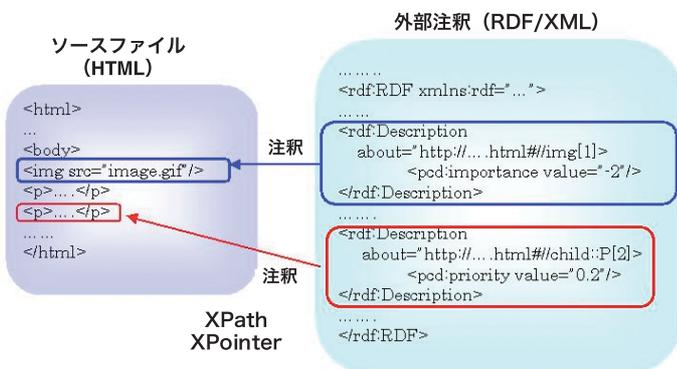


図-3 ソースファイルと外部注釈の対応

タの物理的な量を適応させているので、一度に送られるデータはアクセス環境が快適なほどデータ量が多く、貧弱なほどデータ量が少なくなようなWebデータの配信を行っている。また、画面の特性に合わせたページレイアウトも行われている。

アクセス環境に応じてページのコンテンツとレイアウトを動的に適応化させるための機能として以下の3つが考えられている。

● ページ分割

ページコンテンツを1～2画面分を単位として分割し、続きのコンテンツはリンクをクリックすることでサーバに再接続して受信・閲覧する。

● コンテンツ選択

ナローバンド・狭画面用のコンテンツとブロードバンド・大画面用のコンテンツをあらかじめ用意して環境に応じて配信するコンテンツを選択する。

● 動的コンテンツ生成

既存のコンテンツに付けられた注釈を手がかりに、コンテンツとレイアウトを動的に構成。画像コンテンツは削除・縮小・色数削減などを動的に行う。

Webページのコンテンツとレイアウトをサーバあるいはプロキシで再構成する方法の一例として、これらを実現する外部注釈による技術開発Transcoding Platform<sup>1)</sup>を紹介する。Transcoding PlatformはWebデータを閲覧する環境に合わせて表示体裁やコンテンツの物理的な量を変化させる機能を、ページ作者による注釈記述によって実現している。元のデータはHTML文書を想定しており、注釈はRDFを用いて外部メタデータとして記述され、XPath/XPointerによってXML文書と対応付けられている。Transcoding Platformは、クライアントのHTTP要求を受けてHTTPサーバに応答を求める。アクセス機器に応じて既存のHTML文書から重要度の高いコンテンツだけを抜き出すクリッピングや、携帯電話によるWebアクセスに応じる携帯電話用HTMLに変換するプラグインを組み込んだプロキシである(図-2)。適応化の手がかりとなるのは、HTTPサーバからオリジナルのHTMLファイルに関連づけられた外部注釈ファイルである。外部注釈ファイルは図-3のようにRDFで記述されている。端末として利用される機器の表示能力や通信速度をTranscoding Platformが判別して、コンテンツを動的に変化させることができる。

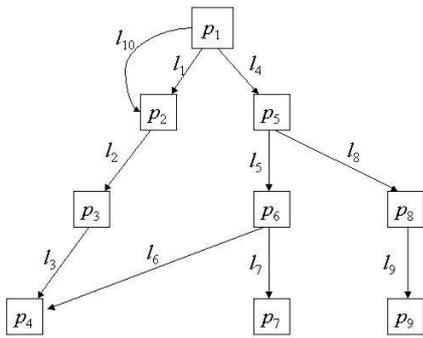


図-4 閲覧順序を意識して構成されたサイトのリンク構造

$\{\} \Rightarrow p_1$	$\{p_1, p_2, p_3, p_5, p_6\} \Rightarrow p_4$
$\{p_1\} \Rightarrow p_2$	$\{p_1, p_5\} \Rightarrow p_6$
$\{p_1\} \Rightarrow p_5$	$\{p_1, p_5\} \Rightarrow p_8$
$\{p_1, p_2\} \Rightarrow p_3$	$\{p_1, p_5, p_6\} \Rightarrow p_7$
$\{p_1, p_2, p_3\} \Rightarrow p_4$	$\{p_1, p_5, p_8\} \Rightarrow p_9$

図-5 ページアクセスのルール

$\{\} \Rightarrow l_1$	$\{\} \Rightarrow l_{10}$
$\{\} \Rightarrow l_4$	$\{l_4\} \Rightarrow l_5$
$\{l_1\} \Rightarrow l_2$	$\{l_4\} \Rightarrow l_8$
$\{l_1, l_2\} \Rightarrow l_3$	$\{l_4, l_5\} \Rightarrow l_7$
$\{l_1, l_2\} \Rightarrow l_6$	$\{l_4, l_8\} \Rightarrow l_9$

図-6 リンク活性化式の例

## アクセスパターンに応じてリンク構造を最適化する適応Webサイト

ある作成者が作成した一連のWebページに対して、ユーザがトップページに最初にアクセスすればページ作成者の設定したリンクに誘導されてリンクを巡行するかもしれない。しかし、ユーザがWeb検索の結果やブックマーク、他のページ作成者が設定したリンクなどを用いてページ作成者のページにアクセスすることが考えられるので、どのページからアクセスされるかは不定である。したがって、ページ作成者がリンク構造の根のページから葉のページまでリンク巡行することを意図してリンクを設定しても、ユーザがそのようにリンク巡行してくれるとは限らず、現状ではこれを期待する以外に方法がない。

Webがユーザにかなりの自由度を与えていることは大

歓迎ではあるが、ページ作成者のページの閲覧やリンク設定の意図を表明する手段がないことは問題である。そこで、ユーザのアクセスパターンに応じてサイトの構造が変化する適応Webサイトが考えられている。

ユーザがページ $p_1, p_2$ を閲覧した後にページ $p_3$ にアクセスしてほしいという意図を $\{p_1, p_2\} \Rightarrow p_3$ と書くことにする。

図-4のサイトに図-5のようなルールを定義することで、ページ $p_1$ からリンクを巡行して各ページを閲覧してほしいというページ作成者の意図が表現できる。これは、リンクをページ対としてモデル化した例である。しかしながら、ページ $p_1$ からページ $p_2$ に2つの異なるリンク $l_1, l_{10}$ が存在する場合は、リンク設定の意図がそれぞれ異なると考えられるので、著者らはリンク巡行権限を動的に変化させる手法としてリンク活性化式を提案している<sup>2)</sup>。リンク活性化式ではユーザがリンク $l_1, l_2$ を共に巡行済みならばリンク $l_3$ を巡行する権限を発行するというルールを $\{l_1, l_2\} \Rightarrow l_3$ と書く。また、リンク $l$ の巡行に何らの制約もないとき $\{\} \Rightarrow l$ と書くことにする。図-5で示したページ作成者の意図をリンク活性化式で表現すると、図-6のようになる<sup>☆1</sup>。

リンク巡行権限を管理してリンク構造を最適化する適応化Webサイトにおいて、権限の管理は制約ととらえられる反面、作者の意図通りにリンク巡行したユーザに特典を与えているという解釈ができる。

リンク活性化式を用いてリンク巡行権限を制御すると、ユーザがページ作成者の意図に反するリンク巡行をしようとしたとき、そのリンクには巡行権限がないと判断され、リンクが巡行できないことになる。このような場合に、リンク $l$ が巡行可能になるためにユーザが巡行済みであるべきリンク集合を求め、これをユーザに提示する機能があれば有用である。このような機能を「リンク活性化式を用いたリンク情報補完機能」と呼ぶ。直観的には、補完すべきリンクの集合は、ユーザが巡行しようとするリンクを $l_i$ 、ユーザ $u$ がこの時点で巡行済みのリンク集合を $L_{navi}(u)$ とすると、左辺が $l$ であるようなリンク活性化式の右辺と $L_{navi}(u)$ との差で求められる。

ページアクセス権限やリンク巡行権限の管理を実際のWebサイトで実現するためには、リンク巡行権限の発行とともにリンクアンカーの有効化、リンク先ページへのアクセス権限の発行を同時に行う必要がある。リンクアンカーの有効化はページコンテンツの動的再構成であるため、次章で紹介する研究と同時進行中である。また、

☆1 リンクをページ対として変換している。図-5の方式と図-6の方式は表現能力が異なるのでまったく同じ意味にはならないが、リンク $l_1$ と $l_{10}$ が巡行済みであることによってリンク巡行が可能なリンクが異なることを表現できている。

リンク情報補完機能はユーザがリンク巡行権限のないリンクを巡行しようとしたことを検知して、ページ作者の意図通りのリンク巡行ができる入り口を動的に生成する必要がある。

## Webコンテンツの個別化 (Personalization) とアクセス管理

Webデータを個々のユーザやグループに適した表現にすることをWebデータの個別化 (パーソナライゼーション) と呼ぶ。商取引の企業—顧客間取引 (B to C) や企業—企業間取引 (B to B) は1対1の個別プロセスで成立しているため、One-to-Oneのサービスが提供できるWebアプリケーションを構築して、Webデータを個別化する必要がある。また、コンテンツの選択やページの見せ方を変化させるだけでなく、アクセス権限を個々のユーザに対して個別に制御できる仕組みも重要である。

一般的なコンテンツ個別化のプロセスを以下に示す。

### (1) ターゲットユーザの特定

ユーザの認証による個人の特定と、どのようなユーザであるかの特定とに分類することができる。顧客—企業間取引やオークションなどの、ある特定の個人に専用のWebビューを提供するアプリケーションではユーザの認証が必要である。一方、ユーザの閲覧方法に基づいてWebビューを変化させるアプリケーションでは、個人の特定ではなく、ある特定の操作を行ったかどうかなどの検証が行われる。

### (2) 個別化を行うためのプロフィールデータの収集

ユーザプロフィールの収集には、ユーザに個人情報や嗜好データを直接尋ねる明示的な方法と、ページ滞在時間、スクロール操作の有無など、利用者の行動を監視することで取得する暗黙的な方法がある。

### (3) ゴールの決定

(2)で収集されたプロフィールデータや履歴データを処理して、Webデータにどのような操作を行うかを決定する工程といえる。操作を決定 (選択) するルールをあらかじめ定義しておく方法や、適合フィードバック技術などを用いてその時々操作を決める方法などが考えられている。

### (4) 中間プロセスの生成

Webビューをどのように生成するかをプログラムする工程であるといえる。実際には、CGI, SSI, MS-ASP, JavaServletなどにおけるモジュール選択あるいはプログラムの動的生成と、スケジューリングを行う。

### (5) 個別化Webビューの生成・配信

(4)で選択・生成されたプロセスを適用してWebビュー

の生成を行い、ユーザに配信する。

ここで、素朴な問題を考えてみよう。それは、どのような場合に個別化を行えばよいか (個別化が必要なのか) という問題である。著者は、各ユーザが他人と違う自分用のデータを見たい場合と、他のユーザと同じデータを見たい場合があり、同様に情報発信者が各ユーザに同一のデータを見せたい場合と、個別のデータを見せたい場合があると考えている。Webがユーザと情報発信者のコミュニケーションである以上は個別に解決されるべき問題である。第0のステップとしてこの問題を考えてほしい。

Webシステムは、一般的に読み出し専用システムであるので、Webアプリケーションで扱うべきアクセス管理は以下のものがある。

#### ● ページのアクセス権限

ページ全体に対する閲覧権限を扱うアクセス権管理であり、アクセス権限があればすべて見ることができ、なければ見るできないようなAll-or-Nothing方式のアクセス管理である。

#### ● リンクのアクセス権限

ページ作成者はページ間の関係を表現するためにリンクを設定する。リンク巡行権限はユーザやグループに対して与えられるが、システムはリンクの巡行が可能かどうかを判別するだけである。

#### ● コンテンツのアクセス権限

ページ中の特定のコンテンツへのアクセス権限を発行する。ページのアクセス権管理と異なり、ページ内でアクセス権限のないコンテンツ以外は見ることができる。

前章でも述べたが、リンクのアクセス権限を管理するためには、リンクアンカーの表示・非表示を制御しなければならない。つまり、リンク巡行権管理はコンテンツのアクセス権管理と同時に以下を考慮する必要がある。

- リンク巡行権限を与えてアンカーに適用
- 巡行権限が与えられたリンクの先のページにアクセス権限を発行

これらのアクセス権管理を実際的な効果を伴って実現するためには、ページデータを変化させなければならない。以上のような機能を実現するために著者らは、ページ作成者自身がルールを設定することでWebデータを動的に変化させる機構ActiveWebを提案している<sup>3)</sup>。そ

ここでは、Webデータを動的に再構成するためにページ作成者が定義するルールを動的再構成ルールと呼ぶ。動的再構成ルールの記法はアクティブデータベースシステムのECAルールを参考にしている。動的再構成ルールはイベント部・条件部・アクション部で構成されている。各部で

**イベント**

- 時刻 (実時刻・仮想時刻)
- 場所 (実位置・仮想位置)
- リンク参照元

**条件**

- ユーザのアクセス履歴
- 利用者グループのアクセス履歴

**アクション**

- コンテンツの追加・削除・置き換え
- アクセス権限の付与

を指定する。

イベント部で時刻と場所が指定できるようにしてあるのは、ユーザが必要とする情報は地理的な位置や時刻によって変化する特性があるからである。たとえば、目的地へのアクセスマップは、地理的に近くから見ている場合と、遠くから見ている場合とでは異なるアクセスマップを表示すべきである。また、ユーザが実際にアクセスする位置や時刻よりも、想定する位置や時刻の方がより重要である場合もある。たとえば、新幹線で東京に向かって最中の位置や時刻よりも、東京駅という目的地(経由地)や到着予定時刻といった仮想的な位置と時刻の方が重要である。そのため、実位置、実時刻ならびに、仮想位置、仮想時刻を考慮している。図-7に動的再構成ルールの設定例を示す。

例1はページdefault.aspに過去5回以上アクセスしているユーザに割引券を発行する簡単な例である。例2は異なるリンク参照元からアクセスすると、同じアンカー文字列で異なるページへのリンクアンカーが追加され有効化される例である。

ActiveWebのシステム概要を図-8に示す。ページ作成者はブラウザのフォームを利用してあらかじめ動的再構成ルールを定義しておく。作成者はブラウザの必要箇所を記入しルールを登録するだけでよい。登録されたルールはRule Generatorによって自動的にXML表現に変換されてサーバに蓄積されていく。一方、利用者がページにアクセスしたとき、Rule Processorは以下の工程で処理を行う。

(1) イベント検知

Rule Processorでは利用者のアクセスを監視しており、アクセスしようとしているページのURLとREADの子

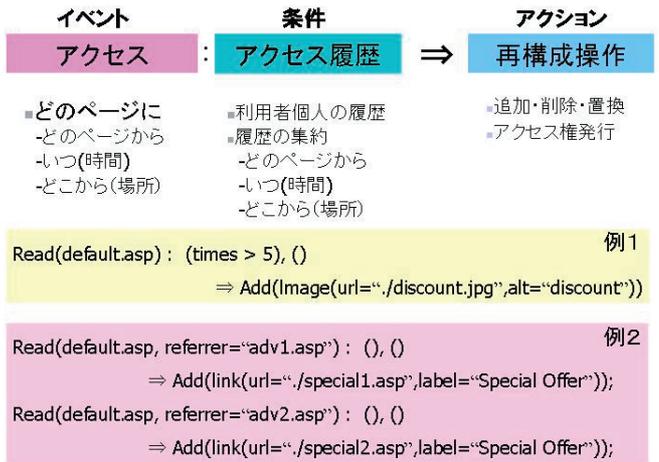


図-7 動的再構成ルールの設定例

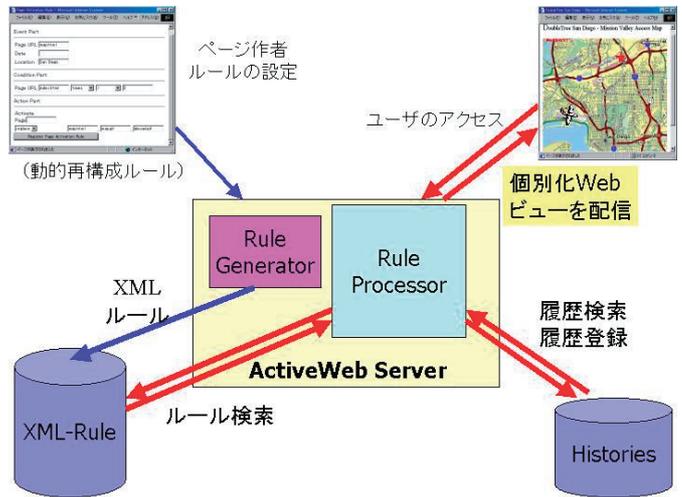


図-8 ActiveWebのシステム概要

要素のURL値を比較し、選択される候補ルールを決める。

(2) ルールの選択

(1)の各候補ルール中でURL要素以外の要素値を満足するルールをトリガ(trigger)する。

(3) 条件の評価

トリガされたルールの条件と履歴情報を比較する。アクセス情報を履歴情報として蓄積領域に格納する。条件を満たすルールに(4)を行う。

(4) アクションの実行順序を決定

(1), (2), (3)で同時に複数のルールが適合した場合、priority属性に従って各ルールのアクションを実行する順序を決定する。

(5) アクションの実行

(4)で決められた順序に従って、ルールのアクションを実行する。テンプレートに従って表示体裁を変更し、

<pre> &lt;!ELEMENT RULES (RULE*)&gt; &lt;!ELEMENT RULE (EVENT,CONDITION?,ACTION)&gt; &lt;!ELEMENT EVENT (READ)&gt; &lt;!ELEMENT CONDITION (USER?,AGGREGATE?)&gt; &lt;!ELEMENT ACTION (ADD HIDE REPLACE ACTIVATE)&gt; &lt;!ELEMENT READ (URL,REFERRER?,     DATE?,LOCATION?,SERVER_DATE?)&gt; &lt;!ELEMENT REFERRER (#PCDATA)&gt; &lt;!ELEMENT DATE (#PCDATA)&gt; &lt;!ELEMENT LOCATION (#PCDATA)&gt; &lt;!ELEMENT SERVER_DATE (#PCDATA)&gt; &lt;!ELEMENT USER (TIMES?,DURATION?,CDT_REFERRER?,     CDT_DATE?,CDT_LOCATION?,CDT_PAGE?)&gt; &lt;!ELEMENT TIMES (#PCDATA)&gt; &lt;!ELEMENT DURATION (#PCDATA)&gt; &lt;!ELEMENT CDT_REFERRER (URL,TIMES)&gt; &lt;!ELEMENT CDT_DATE (DAY,TIMES)&gt; &lt;!ELEMENT CDT_LOCATION (NAME,TIMES)&gt; &lt;!ELEMENT CDT_PAGE (NAME,TIMES)&gt; &lt;!ELEMENT URL (#PCDATA)&gt; </pre>	<pre> &lt;!ELEMENT DAY (#PCDATA)&gt; &lt;!ELEMENT NAME (#PCDATA)&gt; &lt;!ELEMENT AGGREGATE (OLD_INFO?,PAGE?)&gt; &lt;!ELEMENT OLD_INFO (NAME,TIMES)&gt; &lt;!ELEMENT PAGE (NAME,TIMES?,CDT_REFERRER?,     CDT_DATE?,CDT_LOCATION?,CDT_SERVER_DATE?)&gt; &lt;!ELEMENT CDT_SERVER_DATE (DAY,TIMES)&gt; &lt;!ELEMENT ADD (CONTENT)&gt; &lt;!ELEMENT HIDE (CONTENT)&gt; &lt;!ELEMENT REPLACE ((CONTENT),(CONTENT))&gt; &lt;!ELEMENT ACTIVATE (ACTIVATE_PAGE,(true false))&gt; &lt;!ELEMENT ACTIVATE_PAGE (#PCDATA)&gt; &lt;!ELEMENT CONTENT (LINK IMAGE TEXT)&gt; &lt;!ELEMENT LINK (URL,(LABEL IMAGE))&gt; &lt;!ELEMENT IMAGE (URL,ALT?)&gt; &lt;!ELEMENT TEXT (#PCDATA)&gt; &lt;!ELEMENT LABEL (#PCDATA)&gt; &lt;!ELEMENT ALT (#PCDATA)&gt; &lt;!ATTLIST RULE id ID #REQUIRED     priority CDATA #IMPLIED name CDATA #IMPLIED&gt; </pre>
--	---

図-9 ActiveWeb DTD

利用者に再構成されたページを提供する。

動的再構成ルール定義のためのDTDの一部を図-9に示す。

図-10は図-7の例2で示した動的再構成ルールを図-9のActiveWeb DTDにしたがってXML表現した例である。

動的再構成ルールのXML表現を用いてページコンテンツとしてリンクと画像を追加するためのXSLを図-11に示す。

本研究のゴールは、ページ作成者が容易にページ内容と設定したリンク構造の意図を明示的に表明できるWeb情報システムの実現である。そのため、特別な知識を必要としない宣言的なルール記述による個別化手法を提案している。

## 新しいWeb適応とまとめ

著者らも目指していることであるが、文書の電子化という紙の代わりにインターネットではなく、電子化されたデータだからこそ実現できる新しいWebデータの利用が開花こうとしている。多くの研究開発が試みられているが、きわめて実用化の段階に近いものとして灘本ら

のWebデータを受動的に視聴するシステムの技術開発を紹介する<sup>4)</sup>。Webコンテンツと放送コンテンツの融合を目的とし、Webコンテンツを放送コンテンツに変換する機構を研究している。Webコンテンツのタグ構造を解析することにより、図と文章との同期化領域を発見するとともに、キャラクタアニメーションを用いてWebコンテンツをテレビ番組のようなコンテンツに変換している(図-12)。これにより、容易に楽しくかつ受け身でWebコンテンツを取得することが可能となる。このような受動的視聴をさまざまな視聴環境で実現するために演出やストーリーなどのコンテンツ構成を統一的に記述する番組記述言語S-XML(Scripting-XML)の提唱も行っている。

Webデータを利用するツールは一般的にはWebブラウザであるが、近い将来に家電や自動車などの生活に密着した製品もWebデータを積極的に活用するだろう。このように、Web適応の新しい展開とそこからのフィードバックが大いに期待できる。

Webデータを動的に再構成してさまざまなアクセスや要求に適応させるWeb適応技術を紹介した。著者はアクセス機器や通信環境、閲覧形態への適応やWebコンテンツのパーソナライゼーションは、互いに衝突するような概念ではなく、組み合わせて利用することでユーザを快適にするものだと考えている。誌面の都合で割愛し

```

<RULE id="00001">
  <EVENT><READ>
    <URL>default.asp</URL>
    <REFERER>adv1.asp</REFERER>
  </READ></EVENT>
  <ACTION><ADD>
    <CONTENT><LINK>
      <URL>special1.asp</URL>
      <LABEL>SpecialOffer</LABEL>
    </LINK></CONTENT>
  </ADD></ACTION>
</RULE>

<RULE id="00002">
  <EVENT><READ>
    <URL>default.asp</URL>
    <REFERER>adv2.asp</REFERER>
  </READ></EVENT>
  <ACTION><ADD>
    <CONTENT><LINK>
      <URL>special2.asp</URL>
      <LABEL>SpecialOffer</LABEL>
    </LINK></CONTENT>
  </ADD></ACTION>
</RULE>

```

図-10 動的再構成ルールのXML表現

```

<xsl:for-each select="/RULES/RULE/ACTION/ADD">
  <xsl:if test="CONTENT/IMAGE/URL">
    <IMG>
      <xsl:attribute name="SRC">
        <xsl:value-of select="CONTENT/IMAGE/URL"/>
      </xsl:attribute>
      <xsl:attribute name="ALT">
        <xsl:value-of select="CONTENT/IMAGE/ALT"/>
      </xsl:attribute>
    </IMG>
  </xsl:if>
  <xsl:if test="CONTENT/LINK/LABEL">
    <A>
      <xsl:attribute name="HREF">
        <xsl:value-of select="CONTENT/LINK/URL"/>
      </xsl:attribute>
      <xsl:value-of select="CONTENT/LINK/LABEL"/>
    </A>
  </xsl:if>
</xsl:for-each>

```

図-11 リンクと画像をADDするXSL表現

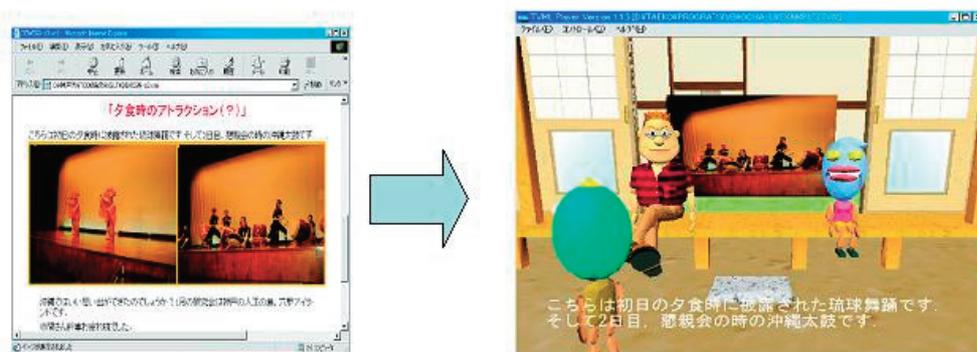


図-12 Webデータの受動的視聴研究

ているが、Webデータの個別化と適応化についてサーベイを行っているので、文献5)も参照いただきたい。

参考文献

- 1) Hori, M., Kondoh, G., Ono, K., Hirose, S. and Singhal, S.: Annotation-Based Web Content Transcoding, <http://www.w9.org/w9cdrom/169/169.html>, WWW9 (1999).
- 2) 清光英成, 田中克己: Webリンクの巡行に基づく動的なリンク活性化とアクセス管理, 情報処理学会論文誌: データベース, Vol.42, No.SIG8 (TOD10), pp.10-20 (2001).

- 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.SIG8 (TOD10), pp.139-147 (2001).
- 4) 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, Vol.42, No.SIG1 (TOD8), pp.103-116 (2001).
- 5) 清光英成, 竹内淳記: Webデータの個別化と環境適応, 情報処理学会論文誌: データベース, Vol.42, No.SIG8 (TOD10), pp.185-194 (2001). (平成15年4月1日受付)

