

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

清光英成[†] 竹内淳記^{††} 田中克己^{†††}

Web データを個々の閲覧者に適した表現にすることを Web データの個別化と呼ぶ。閲覧者個別のページ構成やリンク構造をページ作者が実現するためには Web サーバアプリケーションや CGI プログラミングの知識が必要である。本論文では、アクティブルールを用いた Web ページの動的再構成機構により、ページ作成者が宣言的にルールを記述することで閲覧者個別のページ内容やリンク構造を動的に生成するアイデアを提案する。さらに、アクティブデータベースの ECA ルールと比較して、ActiveWeb システムのルール記述と処理の特徴を議論する。

ActiveWeb: XML-based Reconfiguration for Web Personalization and Access Control

HIDENARI KIYOMITSU,[†] ATSUNORI TAKEUCHI^{††}
and KATSUMI TANAKA^{†††}

In this paper, we propose an idea for using XML-based active rules for deriving Web views and for defining access control by user access behaviors. The major objective of the proposed method is to reflect author (creator)'s intention about his/her Web data and its link structures. In our *ActiveWeb*, the view of a Web page including its hyperlinks is changed according to each user's browsing situation including access history or the aggregated information about all users' access histories. In order to realize the proposed Web-view derivation, first we introduce the notions of *dynamic reconfiguration rules*. Intuitively, each rule is predefined by its author who specifies how to reconfigure his Web pages based on each user's access histories and situations (location and date of accession).

1. はじめに

近年、Web 技術を基盤とした電子商取引が注目されている。特に、顧客-企業間や企業-企業間の商取引を 1 対 1 に行う Web アプリケーションを構築するため、Web データの個別化研究がさかに行われている。

企業だけでなく、個人でホームページを運用しているページ作成者からも各閲覧者に個別のページ構成やリンク構造を提供したいという要求が高まっている。たとえば、ページ作成者のサイトを頻りに訪れる閲覧者に対して、他の閲覧者と異なる情報を提供することで、閲覧者の差別化を図るアプリケーションを構築できれば有用である。しかしながら、Web ページやペー

ジ間のリンクは閲覧者がどのようにアクセスしても変化しない静的なデータである。そのため、閲覧者が過去にアクセスしたページや巡回したリンクなどの情報に基づいてページコンテンツを変化させようとする、内容がほとんど同じ複数の HTML ファイルをあらかじめ用意しておくか、CGI などのサーバアプリケーションを構築しなければならない。そこで、ページ内コンテンツの加工や表示体裁を変化させる機能を容易に実現できるような機構があれば有用だと考えられる。

本論文は、これらの問題を解決するため、アクティブルールを参考にした動的再構成ルールを提案し、Web データを動的に変化させる ActiveWeb を議論する。

2. 本研究の動機と関連研究

2.1 本研究の動機

Web データを個々の閲覧者や閲覧者グループに適した表現にすることを Web データの個別化と呼ぶ。たとえば、閲覧者の嗜好やアクセス履歴から、各閲覧者に個別のコンテンツ構成やリンク構造を提供すること

[†] 神戸大学国際文化学部
Faculty of Cross-Cultural Studies, Kobe University

^{††} 神戸大学大学院自然科学研究科
School of Science and Technology, Kobe University

^{†††} 京都大学大学院情報学研究科
School of Informatics, Kyoto University

が考えられる。

一方、閲覧者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。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。近くから見ている閲覧者には徒歩でのアクセスマップを表示し、遠くから見ている閲覧者には最寄り駅までの交通案内を表示したい。また、閲覧者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、仮想時刻を考慮した Web データの個別化を実現したい。

しかしながら、閲覧者がページにアクセスしたり、リンクを巡行することを考えると、従来の Web システムには以下のような性質がある。

- ページ
ページコンテンツは誰がどのようにアクセスしても変化しない。ページのアクセス権限も.htaccess の記述によって設定されるため、動的に変化するものではない。また、アクセス権限の有無により、ページコンテンツすべてを閲覧できるか、まったくできないかのどちらかである。
- リンク
ページ作成者がリンクを設定すると、設定が更新されるまで誰がどのようにアクセスしても、リンク先は変化しない。
- スタイル
ページの体裁や表現方法はページ作成者が更新を行わない限り変化しない。XML 環境で XSL によってスタイルを変化させることができるが、XSL を記述するためにはある程度の知識が必要である。

Web データを動的に変化させるためには、DynamicHTML や Java, CGI などを使う方法がある。しかしながら、プログラミングの知識を必要とするため、個人でホームページを運営しているページ作成者には構築が困難な場合が多い。そこで、ページコンテンツ、リンク構造、アクセス権限などの Web データを宣言的な言語を定義することで動的に変化させることができる機構があれば有用と考えられる。

2.2 関連研究

提案する ActiveWeb と関連がある主な Web データの個別化研究は、Ceri らと Atzeni らによるものが

ある。

Ceri らは、電子商取引におけるビジネスルールを XML 表現した個別販売のビジネスモデルを提案している^{2),3)}。利用者プロフィールデータは ID, 電子メールアドレス, アクセス履歴, 購買履歴などから構成されている。また、ビジネスルールをアクティブデータベースシステムの ECA ルールを模した XML 記法を用いて実現しようとしている。ここでは、定義されたビジネスルールを用いて、電子商取引業者が顧客に推薦する商品を動的に決定するためのツールの構築を目的としている。Ceri らのプロファイルは、利用者に個人情報を直接尋ねて収集するとともに、ページの閲覧やページ間の移動を追跡し、さらに購買履歴もプロフィールデータとして記憶しておくという特徴を持っている。Ceri らの研究と ActiveWeb の主な差異は以下のとおりである。

- Ceri らは、アクティブルールを利用者プロフィールデータの更新に利用しているが、ActiveWeb では、Web データを変化させるためのルールとして利用している。
- Ceri らの研究は、プロフィールを利用した閲覧者サイドの個別化アプローチであるのに対して、ActiveWeb は Web データの個別化に関するページ作者の意図をルールを用いて記述するため、ページ作者サイドのアプローチといえる。
- ActiveWeb は、閲覧者のアクセス時の地理的・時間的な状況を考慮しているが、Ceri らは履歴としての時刻しか利用していない。

Atzeni らは、利用者プロフィールを用いて、利用者が興味を持っていると想定されるページを集めた仮想サイトを生成する ARANEUS システムを提案している¹⁾。Atzeni らの研究と ActiveWeb の主な差異は以下のとおりである。

- ARANEUS では、サイトの構造を個別化するため、ページコンテンツやスタイルを変化させないが、ActiveWeb では、アクセス権限やページ内の一部のコンテンツ、リンク構造などを変化させることができる。
- Atzeni らの研究は、利用者の要求を満たすサイトを構築することを目的としているが、我々はページ作成者の意図を Web データの個別化に反映することを目的としている。

また、これらの研究では、Web データを変化させる仕組みはプログラム内に埋め込まれている。我々の提案する ActiveWeb は、ページ作成者が Web データをどのように変化させたいかという意図をルールとして

宣言的に定義できることを目的としている．さらに，閲覧者がコンテンツを利用するための制約という観点からでなく，ページ作成者の意図どおりの閲覧やリンク巡行に対して，閲覧者にもメリットがあるハイパーテキストシステムの構築を目指している点でこれらの研究と異なる．

Web データの環境適応に関する関連研究では，Horiらと北山らによるものがある．

Horiらは，Web データを閲覧する機器や通信環境にあ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構を，ページ作者による注釈記述によって実現している^{6),14)}．北山らは，端末の能力に応じて Web データを生成するビジネスアプリケーション構築機構 Dharma を開発している^{5),7)}．

Horiらや北山らは，閲覧者が利用している環境の物理的側面に着目した Web データの環境適応を目的としているのに対して，ActiveWeb は，閲覧者がアクセスしている時刻や場所といった，閲覧者がおかれている状況に着目して Web データを適応させることを目的としている．

3. Web データの動的再構成

3.1 Web データのアクセス管理とコンテンツの再構成

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
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，なければ見ることができないような All-or-Nothing 方式のアクセス管理である．システムはユーザやグループを対象にパスワード認証などからページのアクセス権限を発行する．
- リンクのアクセス権
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．リンク巡行権限は閲覧者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
ページのアクセス権管理と異なり，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．そのため，ページ作成者が主張する意図の違いを以下のような視覚的表現で実現可能である．
 - － コンテンツの配置が違う．
 - － 表示されるコンテンツ自体が違う．

- － 文字列中のある単語がすべてリンクアンカーになる．
- － アンカー文字列が画像バナーになる．
- － 同じアンカーをクリックしてもリンク先ページが違う．

以上のような機能を実現するために，本論文では，ページ作成者自身がルールを設定することでページコンテンツを動的に変化させる機構を提案する．また，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，リンク巡行のアクセス権管理は，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ともなっ実現するためには，ページデータを変化させなければならない．そこで，Web データを動的に再構成するための動的再構成ルールを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記述する．また，動的再構成ルールを XML 形式で表現する．

3.2 動的再構成ルール

動的再構成ルールは以下の記法で表現する．

イベント部: 履歴条件部 ⇒ アクション部

ルール中の各部には，

- イベント部
特定のページアクセス，リンク巡行，アクセスしている位置・時刻などの閲覧者のアクセス時の行為．アクセスページの URL の記述は必須である．
- 履歴条件部
アクセス回数，最後にアクセスした時刻などの閲覧者履歴とサーバに蓄積されている全アクセス履歴から導出される集約情報に対する条件．利用者個別の履歴と集約情報は分けて記述する．
- アクション部
ルールを満たしたときに発火するアクション．追加 (add), 隠蔽 (hide), 置換 (replace), アクセス権発行 (activate) を用意している．

を記述する．

動的再構成ルールの記述例を以下に示す．

【例 1】閲覧者がページ default.asp にアクセス要求をしたときに，リンク元ページが adv1.asp であれば，special1.asp へのリンクを追加し，リンク元ページが adv2.asp であれば，special2.asp へのリンクを追加するルールは，図 1 のように記述する．

```
read(default.asp, referrer="adv1.asp"): (),()
⇒ add(link="special1.asp", label="SpecialOffer")
read(default.asp, referrer="adv2.asp"): (),()
⇒ add(link="special2.asp", label="SpecialOffer")
```

図1 【例1】参照元ページによりリンク先を変化させる例

Fig. 1 Example 1: Changing destination anchor of a link based on the source anchor.

```
read(map.asp, location="kobe"): (),()
⇒ replace(image="airport.gif", image="station.gif")
```

図2 【例2】アクセス位置により画像を置換する例

Fig. 2 Example 2: Replacing figures based on location.

```
read(default.asp): (page="copyright.asp", times > 4),()
⇒ hide(link="copyright.asp")
```

図3 【例3】アクセス回数によりリンクアンカーを隠蔽する例

Fig. 3 Example 3: Hiding a link anchor based on times of accession.

```
read(new_year.asp, date="2001/1/1"): (page="xmas.asp", times > 2),
(page="xmas.asp", times > 1000, location="kobe")
⇒ activate(page="new_year.asp", "true")
```

図4 【例4】アクセス履歴によりページアクセス権限を付与する例

Fig. 4 Example 4: Issuing access right to the page based on access history.

【例2】閲覧者が神戸市内から map.asp を閲覧する場合に、station.gif を表示し、それ以外の場合は airport.gif を表示するルールは、図2のように記述する。

【例3】閲覧者が過去にページ copyright.asp を5回以上アクセスしていれば、ページ default.asp から copyright.asp へのリンクを隠蔽するルールは、図3のように記述する。

【例4】閲覧者が過去に3回以上ページ xmas.asp にアクセスし、神戸市内から xmas.asp への全アクセス数が1000より多いなら、2001年1月1日の new_year.asp へのアクセス権限を閲覧者に発行する場合、図4のように記述する。

図1～図4では、アクセス要求をするページのURLや利用者のアクセス時の状況と、利用者個別の履歴条件・集約情報の条件を設定したあと、条件を満たした場合の再構成操作を記述している。特に、referrer, date, location はイベント部では read 要求を修飾する変更子 (modifier), 履歴条件部ではアクセス履歴を修飾する変更子としての意味を持つ。

イベント部や履歴条件部で用いる条件は、‘神戸’や‘13:00’といったインスタンスを直接記述だけでなく、‘近畿地方’や‘午後’というようなメタな記述

も可能である。たとえば、‘京都’・‘大阪’・‘神戸’という異なる場所であっても、再構成操作の内容が同じである場合は、‘近畿地方’というメタな記述をすることで同じようなルールを何度も書く手間が省ける。記法は省略するが、このように記述する場合は、あらかじめ‘近畿地方’は‘京都’・‘大阪’・‘神戸’を含むという semantics を定義すればよい。

また、アクセス権を制御するために、作者はページごとに標準のアクセス権を記述する。アクセス権はそれぞれページごとに記述する。標準のアクセス権を記述しない場合は、誰でもアクセスすることができる“true”となる。図4の例では、作者はページ new_year.asp のアクセス権を“false”としておき、例4の条件を満たした利用者だけが new_year.asp のアクセス権を発行され、アクセスすることができる。

4. XML を用いた動的再構成手法

Web ページの個別化は、ASP や CGI だけで実装できるが、ページごとにプログラムを用意しなければならないことが多く、効率的であるとはいえない。また、現状のページの動的再構成はこれらの専門的知識が不可欠であるような手続き的手法である。そこで本章では、Web ページの表現を閲覧者個別に変化させ

```

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

```

図 5 ActiveWeb DTD

Fig. 5 ActiveWeb DTD.

る宣言的手法を提案する。動的再構成ルールの XML 表現を用い、条件を評価した後、スタイルシートを用いて表示体裁を変更している。

4.1 動的再構成ルールの XML 表現

動的再構成ルールを XML データとして記述するため、まず、動的再構成ルールを XML 実現にするための ActiveWeb DTD を図 5 に示す。これにより、図 1 の例は図 6 の XML で、図 4 の例は、図 7 の XML でそれぞれ表現することができる。

Web システムが扱うのは読み出し専用データであるため、閲覧者がサーバに発行できるのは読み出し要求だけである。そのため EVENT の子要素は READ であり、READ の子要素には、閲覧者が読み出しを要求するページの URL、参照元、アクセス時刻・場所を型定義している。

```

<RULE id="00001">
<EVENT><READ>
<URL>default.asp</URL>
<REFERRER>adv1.asp</REFERRER>
</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 6 図 1 の XML 表現

Fig. 6 XML representation of Fig. 1.

```

<RULE id="00005">
<EVENT><READ>
<URL>newyear.asp</URL>
<DATE>2001/1/1</DATE>
</READ></EVENT>
<CONDITION><USER><CDT_PAGE>
<NAME>xmas.asp<NAME>
<TIMES>more_than, 2</TIMES>
</CDT_PAGE></USER>
<AGGREGATE><PAGE>
<NAME>xmas.asp</NAME>
<TIMES>more_than, 1000</TIMES>
<CDT_LOCATION>kobe</CDT_LOCATION>
</PAGE></AGGREGATE></CONDITION>
<ACTION><ACTIVATE><ACTIVATE_PAGE>
newyear.asp, true</ACTIVATE_PAGE>
</ACTIVATE></ACTION>
</RULE>

```

図 7 図 4 の XML 表現

Fig. 7 XML representation of Fig. 4.

条件部分は、アクティブデータベースシステムと同じく、定義してもしなくてもよい Option となっている。イベント部と類似の型定義をしているが、イベント部の要素はアクセス時の状況であり、履歴条件部の要素は履歴中に存在する過去のアクセス時の情報を示している。

アクション部分は、コンテンツの追加 (add)・隠蔽 (hide)・置換 (replace) やページアクセス権発行 (activate) を記述する。コンテンツに対する操作は、子要素としてリンク・画像などの Web ページに表現

可能なコンテンツを記述できるようになっている。特に、置換 (replace) は、アンカー文字列を画像パナーに置き換えたりするような、メディアの変化に追従できる型定義としている。

また、ルールには属性が存在する。id はシステムがルールを識別するために用いられるため、サイト中で唯一である。id は定義されたルールを格納する際にシステムによって自動的に付与される。name はページ作者がルールの作成時に付与する。作者がルールを識別するために用いるため、サイト中で唯一でなくてもかまわない。priority は、システムによってルール間の競合の可能性を指摘されたルールに対してのみ作者が設定する。ルールの競合は、作者によって複数のルールが定義された場合に、操作の内容によっては対象となるコンテンツやページがすでに別のルールによって状態が変更されており、作者の意図したアクション部を実行することができない問題をさす。ここで述べる状態とは、コンテンツは存在の有無であり、ページはアクセス権の有無のことである。

ページコンテンツに対する操作のうち、競合を起こす可能性がある操作は隠蔽と置換である。隠蔽や置換の引数となっているコンテンツが複数のルールで操作の対象となっている場合、利用者のアクセス状況やアクセス履歴によっては競合を起こすため、どのルールを実行すべきかを明確に記述しておく必要がある。また、ページのアクセス権を操作する Activate は、コンテンツに対する他の操作よりも優先的に実行されるが、複数のルールで同一 URL のページに関するアクセス権の操作を行う場合には競合する可能性がある。

そこで、ページ作者はルールを作成する際に、他のルールと競合を起こす可能性を持つルールに対してのみ、ルールの優先度 (priority) を記述する。優先度は、どのような条件のアクセスであっても、作成したルールと表示されるコンテンツやページの間で矛盾を発生させないようにする。そのため、id 属性を用いて記述し、たとえシステムによって条件を満たすルールが多数選択されても、作者の意図したとりの内容が利用者に対して提供される。

4.2 動的再構成ルールの評価と実行

まず、本手法の概要を図 8 に示す。ActiveWeb の処理はすべてサーバで行われる。

ページ作者はブラウザの FORM 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

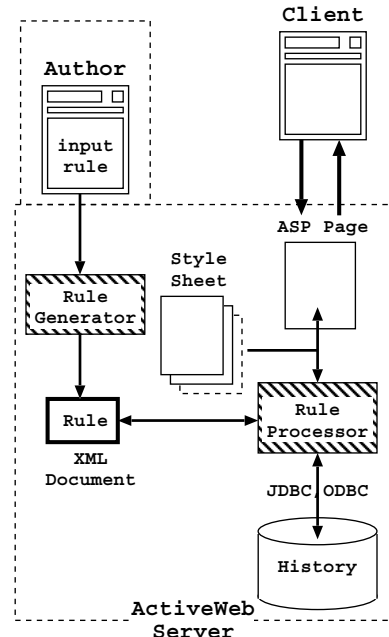


図 8 システムの概要

Fig. 8 Outline of prototype.

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、利用者に再構成されたページを提供する。

図 6 の例では、ルールの id が “00001”, “00002” である動的再構成ルールのどちらを適用するかで、表示すべきアンカー文字列とリンク先が決定する。どち

```

<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 9 XSL による ADD の記述例

Fig. 9 XSL representation of a function ADD.

らのイベントでもない場合は 2 つとも適用されないの
で、動的再構成処理は行われない。アクションに記述
された動的再構成処理は、ADD のみスタイルファイ
ルで実行される。アンカー文字列やバナー画像とリン
ク先 URL とを一般的なリンクとして閲覧者に提供す
るために、図 9 のようなテンプレート記述を適用す
る。HIDE と REPLACE は Rule Processor により、
該当する文字列やタグ領域に対してコンテンツの操作
を行う。

5. アクティブデータベースシステムとの差異

データベースシステムは、更新トランザクション
と読み出しトランザクションの両方を扱う。一般的に
Web システムは読み出し専用のアプリケーションで
あるため、閲覧者の読み出し要求に対する操作のみを
行う。ページコンテンツの動的再構成は、閲覧者がア
クセスするページの見え方を変化させるため、デー
タベースのビューインスタンスを生成する操作とよく似
ている。

本章では、アクティブデータベースシステムのルー
ルと ActiveWeb のルールを、記述内容やルール評価
とアクション起動のタイミングについて比較する。

5.1 知識モデル

アクティブデータベースのアクティブルールはイベ
ント・条件・アクションの組である。我々の ActiveWeb
もアクティブルールに準じた分類で表 1 に整理した。

アクティブデータベースシステムではイベントに構
造操作・トランザクション・例外などの組合せを記述
できる。たとえば、テーブルにタプルが挿入される

表 1 ルールの記述内容

Table 1 Comparison between Active DB and ActiveWeb
on rule expressions.

	Active DB	ActiveWeb
Event	構造操作 トランザクション 例外	ページアクセス 閲覧者の位置 時刻
Condition	データベースの状態	閲覧者個別履歴 履歴情報の集約
Action	データベースの操作	コンテンツの 再構成、 アクセス権発行

トランザクションが中断した事象を条件部分の評価
を開始する引金 (trigger) として記述することができ
る。一方、我々の ActiveWeb は、閲覧者の位置や時
刻などの閲覧者の状態とアクセスされるページの組を
記述し、指定されたページにアクセスされたときに条
件部分の評価してアクションを適用する。特に、イベ
ントは Web データの Read のみを監視しているため、
あるルールのアクションが他のルールのイベントを引
き起こすことによるアクティブルールのループ問題は
起こらない。条件部分は、アクティブルールでは指定
したトランザクションが開始されたかや、他のルールの
アクションが実行中かなどのデータベースの状態を
記述する。アクション部分はともに操作を記述してい
る。特に、*replace* はアクティブデータベースの *do
instead*⁴⁾ とよく似た機能である。

5.2 実行モデル

アクティブルールの条件は、イベントが発生したら
即座に評価されるものと、ある期間だけ遅延して評価
されるものがある。ActiveWeb では、閲覧者の read
要求に対してページコンテンツを動的に変化させるた
めにルールを適用するため、イベントによるルール選
択と条件は即座に評価されサーバがアクセス情報を記
録する。データベースシステムでは、条件の評価対象
がトランザクションの実行状況やその時点でのデー
タベースの状態であるのに対して、ActiveWeb システ
ムでは閲覧者個別のアクセス履歴や全閲覧者のアク
セス履歴の集約などが評価の対象となる点で異なる。
アクションにも、即時起動型 (immediate) と遅延起動
型 (deferred) があるが、アクティブデータベースで
はルールでデータベースの更新処理を記述できるため、
一度しか起動しないアクションや一度条件を満たせば
その後はグループ内のすべてのトランザクションに対
して適用できるアクションを記述できるようになって
いる。ActiveWeb も、即時起動型と「来月から画像
を置き換える」などの遅延起動型のアクションが必要

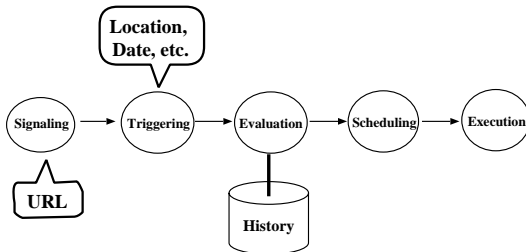


図 10 ルールの選択とアクションの実行
Fig. 10 Trigger and execution of rules.

表 2 実行モデルの比較

Table 2 Comparison between Active DB and ActiveWeb on execution model.

	Active DB	ActiveWeb
Conditon	Deferred	Immediate
	Immediate	
	Detached	
Action	Immediate	Immediate Deferred
	Deferred	
	Detached	

である。これに加えて「明日から三カ月、夏用のスタイルにする」などの条件はルールで指定することも考えられるため、ある一定期間だけアクションを適用する機能も必要である。

アクティブデータベースシステムではアクションの起動タイミングをイベントとなったトランザクションのコミットに依存させることができる。ActiveWebシステムにはこの機能は必要ない。

ActiveWeb システムの動的再構成ルールの処理は以下のように行われる(図 10)。

- イベント発生を検知 (Signaling)
- 検知されたイベントが記述されているルールを選択 (Triggering)
- 選択されたルールの条件を評価 (Evaluation)
- 条件を満たすルールのアクションの実行方法を決定 (Scheduling)
- アクションの実行 (Execution)

実行モデルではルール処理に違いがないように見えるが、ActiveWeb システムは、ページの元データに変更を加えることがないので、スケジューリング工程でルールの衝突だけを監視すればよい。たとえば、同一のイベントにより選択された複数のルール中で条件部分を持たないルールと条件を満たしてアクションを実行しようとしているルール間の適用順序の決定がこれにあたる(表 2)。

6. ま と め

Web システムにおけるアクセス管理を具体的に実現するために、ページコンテンツの動的再構成について議論した。動的再構成ルールをアクティブデータベースシステムの ECA ルールにならって記述してルール処理を行い、閲覧者個別のページビューを表現する実装法を示している。特に、閲覧者のアクセスする位置や時刻を動的再構成ルールのイベント部と履歴条件部に変更子(modifier)として記述できる。また、アクティブデータベースの知識モデル記述と動的再構成ルールによる表現の違いを議論するとともに、実行モデルにおける差異にも言及している。

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表現できる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実現手法として宣言的なルール記述による、ページ表現の個別化を提案した。実装したプロトタイプシステムでは、コンテンツがページ内のどの場所に追加されるかを指定できない。Web データが頻繁に更新されるという現実を考慮すればこの機能を実現するためには、ルール定義とエディタの連携を必要とする。また、ページ作者の意図と利用者の要求を調整する Mediator 機能や、複数のページ作者の意図や主張の調整を行う Coordinator 機能も必要と考える。さらに、ブラウザのキャッシュやプロキシの使用によって、ページの見え方が変化しない問題がある。ActiveWeb では、Cookie に利用者のアクセス履歴を格納しているため、毎回サーバにアクセスするが、Cookie を拒絶するようなプロキシではこの問題が起こる。個別化されたデータを確実に閲覧者に配送できる機構も必要である。本論文では特に言及しなかったが、ともに今後の課題である。

謝辞 本研究をすすめるにあたり、多くのご意見と助言をいただいた本学工学部ならびに自然科学研究科田中研究室の皆様へ深く感謝いたします。また、本研究の一部は、日本学術振興会未来開拓学術推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(プロジェクト番号 JSPS-RFTF97P00501)ならびに、文部省科研費「分散型ハイパーメディアからの構造発見とアクセス管理」(課題番号 12680416)による。ここに記して謝意を表します。

参 考 文 献

- 1) Atzeni, P., Mocca, G. and Merialdo, P.: To

- Weave the Web, *Proc. VLDB'97*, pp.206-215 (1997).
- 2) Ceri, S., Fraternali, P., Maurino, A. and Paraboschi, S.: One-to-One Personalization of Data-Intensive Web Sites, *Proc. WebDB'99*, pp.1-6 (1999).
 - 3) Ceri, S., Fraternali, P. and Paraboschi, S.: Data-Driven, One-To-One Web Site Generation for Data-Intensive Applications, *Proc. VLDB'99*, pp.615-626 (1999).
 - 4) Dittrich, K.R., Gatzju, S. and Geppert, A.: The Active Database Management System Manifesto: A Rulebase of ADBMS Feature, *Rules in Database System (LNCS-985)*, pp.1-17, Springer (1995).
 - 5) 広瀬紳一, 北山文彦, 久世和資: 多種端末向け Web アプリケーション構築システム Dharma : ビューオブジェクト生成と HTML 生成機構, 情報処理学会第 57 回全国大会文集分冊 3, pp.392-393 (1998).
 - 6) Hori, M., Kondoh, G., Ono, K., Hirose, S. and Singhal, S.: Annotation-Based Web Content Transcoding, *Proc. WWW9*, pp.197-211 (1999).
 - 7) 北山文彦, 広瀬紳一, 久世和資: 多種端末向け Web アプリケーション構築システム Dharma : システム概要とアプリケーションオブジェクト, 情報処理学会第 57 回全国大会文集分冊 3, pp.390-391 (1998).
 - 8) Kiyomitsu, H., Takeuchi, A. and Tanaka, K.: ActiveWeb: XML-Based Active Rules for Web View Derivations and Access Control, *Proc. the Workshop on Information Technology for Virtual Enterprises (ITVE 2001)*, Orlowska, M. and Yoshikawa, M.(Eds.), Australian Computer Science Communication, Vol.23, No.6, pp.31-39, IEEE Press (2001).
 - 9) Kiyomitsu, H., Takeuchi, A. and Tanaka, K.: Web Reconfiguration by Spatio-Temporal Page Personalization Rules Based on Access Histories, *Proc. Symposium on Applications and the Internet (SAINT 2001)*, Ikeda, T. and Douglass, F.(Eds.), IEEE-CS/IPSJ, pp.75-82, IEEE Press (2001).
 - 10) Kiyomitsu, H. and Tanaka, K.: Navigation-Dependent Web-Views: Changing Web Pages and Link Structures By User Navigation History, *Proc. ICSC '99*, Vol.LNCS-1749, pp.421-426 (1999).
 - 11) 清光英成, 竹内淳記, 田中克己: アクセス履歴と利用者の位置に基づくコンテンツの再構成, 情報処理学会研究報告, Vol.2000, No.3, pp.17-24 (2000).
 - 12) 清光英成, 竹内淳記, 田中克己: 閲覧履歴に依存した Web コンテンツの動的生成方式, DEWS2000 論文集 CD-ROM (2000).
 - 13) 竹内淳記, 清光英成, 田中克己: アクティブルールに基づく Web 個別化・環境適応システム: ActiveWeb の実装, DEWS2001 論文集 CD-ROM (掲載予定) (2001).
 - 14) IBM 東京基礎研究所: XML オーサリングツール構築技術: FREEDOM. <http://www.trl.ibm.co.jp/projects/freedom/>

(平成 12 年 12 月 20 日受付)

(平成 13 年 4 月 10 日採録)

(担当編集委員 河野 浩之)



清光 英成 (正会員)

1994 年図書館情報大学図書館情報学部卒業. 1996 年同大学大学院図書館情報学研究科図書館情報学専攻修士課程修了. 1998 年奈良先端科学技術大学院大学情報科学研究科博士後期課程中退. 同年神戸大学経済学部助手. 2001 年同大学国際文化学部講師, 現在に至る. データベースシステム研究, Web 情報システム研究, Web データの個別化・環境適応研究に従事. IEEE Computer Society, ACM 各会員.



竹内 淳記 (学生会員)

2000 年神戸大学工学部情報知能工学科卒業. 現在同大学大学院自然科学研究科修士課程在学中. Web 情報システム研究, Web データの個別化・環境適応研究に従事.



田中 克己 (正会員)

1974 年京都大学工学部情報工学科卒業. 1976 年同大学大学院修士課程修了. 1979 年神戸大学教養部助手, 1986 年同大学工学部助教授. 1994 年同大学工学部教授 (情報知能工学科) 1995 年同大学大学院自然科学研究科専任教授 (情報メディア学専攻). 2001 年京都大学大学院情報学研究科教授, 現在に至る. 工学博士. 主にデータベースの研究に従事. 人工知能学会, 日本ソフトウェア科学会, IEEE Computer Society, ACM 等会員.