

アクセス履歴の集約とメタルールに基づく Web コンテンツの アクセス管理機構

竹内 淳 記[†] 清光 英 成^{††} 田中 克 己[†]

本研究は、インターネット利用者のページアクセス履歴に基づいてページの内容を動的に変化させることを目的とし、ページに表示するコンテンツを変化させるための動的再構成ルールについて議論する。動的再構成ルールはページ作者が定義し、ページアクセス時の情報やアクセス履歴とコンテンツの表示や隠蔽との依存関係を記述する。さらに、動的再構成ルールを操作するためのメタルールを議論する。メタルールは、動的再構成ルールの関係や禁則などを記述するルールである。これら2つのルールによって、ページ作者は履歴に基づいてページコンテンツを変化させることができ、利用者ごとに異なった内容のページを提供することができる。

Access Control of Web Content Based on Access Histories, Aggregations and Meta-Rules

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

In this paper, we propose an idea for changing web pages based on a user's access history. The major objective of proposed methodology is to reflect a web page to an author's intention using dynamic reconfiguration rules. A dynamic reconfiguration rule is usually defined by an author. An author can provide appropriate web content using some rules described in a declarative manner. And more, we discuss about meta-rule to manage relationship of dynamic reconfiguration rules.

1. はじめに

インターネット利用者が Web ページ作成用のソフトウェアを用いて手軽にページを作成することができるようになってきている。さらに、Web ページ作成用のツールの発達により、実際に HTML を記述してページを作成することが少なくなってきた。特に JavaScript や DynamicHTML の自動生成機能などを用いてページコンテンツが異なる動的なページが増えてきている。

しかしながら、どのような閲覧の仕方をしてもページに表示されるコンテンツは同じであるといった問題がある。例えば、商用サイト等で利用されているバナー広告は利用者にクリックしてもらうことが目的であるが、バナー広告をクリックしても利用者には利

点が少ないため、バナー広告が効果的に利用されていないといった問題がある。そこで、実際にバナー広告をクリックしてページにアクセスした利用者には、直接 URL を入力した利用者とは異なり、ページコンテンツが変わるといったサービスを提供できるようなシステムが必要である。しかし、このようなシステムを導入するには、CGI や ASP^{*} といったプログラミング等の特別な知識を必要とするため、初心者が動的に異なったコンテンツを表示するページを作成することは困難である場合が多い。

また、現在の Web ページには、例えば、利用者にパスワードを発行してページのアクセス権を与えるといったようなページ単位でアクセス権を発行しているページが多数存在する。しかしながら、このようにページ単位でアクセス権を与える/与えないといった All-or-Nothing 的なアクセス管理だけでは、個々の利用者ごとに提供できるコンテンツが限られてしまうといった問題がある。

[†] 神戸大学大学院自然科学研究科
Graduate School of Science and Technology, Kobe University

^{††} 神戸大学大学院経済学研究科
Graduate School of Economics, Kobe University

^{*} Microsoft Active Server Page

筆者らはこのような問題に対して、ページ単位だけでなくコンテンツ単位で利用者ごとに閲覧可能とするアクセス権の発行を議論してきた^{1),2)}。同時に、インターネット初心者へのページ作成を支援するアイデアも提案してきた。具体的には、ページ作者がアクセスする利用者に対して、個々の利用者の履歴情報やサーバに蓄積した全利用者に対するコンテンツの表示履歴から動的にページコンテンツを変化させるために、宣言的に記述することができる動的再構成ルールを提案した。そして、そのルールの記述はプログラミングの知識を持たないページ作者でも容易に記述することができた。

しかし、ページ作者がこのような宣言的なルールを好きなように記述していると、コンテンツをページに表示する際にルール間で衝突を起こすような記述をする可能性がある。そのようなことがあってはページ作成を支援していることにはならない。利用者の履歴に基づいてページに表示するコンテンツを容易に記述できるルールだけでなく、そのルール間の関係を記述できるように機構が合わせて必要になると思われる。また、同時に複数のルールを変化するといったようなルールに対する操作を行なう支援もページ作成には必要である。

そこで本論文は、ページに表示するコンテンツを動的に変化させるルールである動的再構成ルールを管理するためにメタルールを提案する。メタルールはページ作者が定義し、動的再構成ルールに対して操作を行なうルールである。この2種類のルールは、アクティブデータベース³⁾に用いられるECAルールをベースに考える。また、ページ利用者個別のアクセス履歴だけでなく、利用者全体に対して実行したルールの履歴やページアクセスの総数、どの時間帯にアクセスが集中しているかなど、サーバに蓄積した全てのアクセス履歴の中からページ作者が選出した集約情報に用いてルールを記述し、そのルールに基づいてコンテンツをページに表示するといったことについても議論する。

2. 動的再構成ルール

動的再構成ルールとはページ作者が宣言的に記述できるルールであり、動的にページコンテンツを変化させるためにアクセス時の利用者の閲覧状況や利用者ごとのページアクセス履歴、さらにサーバに蓄積される利用者全体のアクセス履歴を利用する。

本論文では、利用者個別のページアクセス履歴を利用者履歴とし、全利用者に対してサーバが実行した

ルールの蓄積情報をルール実行履歴とする。

2.1 利用者ごとのイベント情報

利用者のページアクセス時に得ることができる情報をイベント情報とする。ページ作者はイベント情報によって異なるコンテンツを提供することができる。イベント情報の属性として、

- date：時刻
- location：位置
- referer：参照元

を用意した。サーバはこれらの情報を各々の利用者のアクセスごとに獲得し、そのデータはページアクセス履歴情報として全て蓄積していく。

2.2 ページアクセス履歴

サーバに蓄積されたページアクセスの履歴には、各々の利用者だけのページアクセス履歴と、ページ利用者全体のアクセス履歴と2種類存在する。

2.2.1 利用者履歴

利用者個別のアクセス履歴である利用者履歴の属性として、

- date：時刻
- location：位置
- referer：参照元
- times：アクセス回数
- old_info：同一 URL の最後のアクセス時に実行したルール名

を用意した。前3つの属性は、イベント情報を蓄積していったデータであり、times や old_info といった属性は利用者のアクセス時に更新されていくデータである。

動的再構成ルールでは、イベント情報やページアクセス履歴の属性と属性の条件を満たした場合に実行される関数および内容の依存関係を記述する。そして、依存関係を記述したルールにはページ作者がそれぞれ異なるルール名を便宜上記述しておく。このルール名は利用者履歴の属性である old_info や後に述べるルール実行履歴の属性の rule で利用される。また、サーバでは記述されたルールごとに unique な ID をつけておき、ページ作者がルールに同じ名前をつけてしまった場合に、ルールを一意に判別できるようにする。

2.2.2 アクセス履歴の集約

利用者全体のアクセス履歴をサーバに蓄積し、集まった全てのアクセス履歴のデータの中からページ作者は必要と思われる情報を条件を与えて選出すること

ができる。ページ作者は、利用者全体が Web ページをどのように閲覧しているかを把握することができる。

利用者全体のアクセス履歴の属性には、

- アクセス総数
- アクセスの時間帯
- 参照されているページの上位 5 件

などが挙げられる。また、サーバで利用者のアクセス時に実行される動的再構成ルールを監視し実行したルール名を蓄積することで、全利用者に対して実行したルールの履歴を得ることができる。このルール実行履歴は利用者全体に対するアクセス履歴の一部である。

ルール実行履歴に基づいた動的再構成ルールを記述するために属性として、

- rule : 実行したルール名
- combination : 実行したルール名の組合せ
- times : ルールを実行した回数

の 3 つを用意した。

利用者履歴やルール実行履歴のようなページアクセス履歴に基づいた動的再構成ルールにもページ作者はルール名をつけておき、ルールが実行されたときにはアクセス履歴として蓄積していく。

2.3 動的再構成ルールの記述方法

イベント情報や利用者履歴、ルール実行履歴などのアクセス履歴を用いてページに表示されるコンテンツを変化させるために、コンテンツの動的再構成ルールの記述方法を以下に示す。ここで記述を簡単にするために、ECA ルールと同様の形式を採用し本論文では以下のように定義する。

```
page_name : incidental,histories => function
```

この式の各要素は、

- page_name
動的再構成ルールを適用するページ名
- incidental : (date,location,referrer)
date,location,referrer といったページアクセス時に獲得できるイベント情報
- histories : (times,old_info)
または、
histories : (rule(combination),times)
利用者履歴、またはルール実行履歴などサーバに蓄積されているアクセス履歴情報に対する条件
- function : add,hide,replace
ルールの条件を満たしたときに実行するアクションを表わしている。function には、add(追加)、hide(隠蔽)、replace(置換)の 3 つを用意した。例えば、利用

者がページ index.html にアクセス要求をしたときに、参照元のページが root.html であれば、ページに表示するコンテンツ before.jpg を after.jpg に置き換える場合は以下のように記述する。

```
index.html : (referer == root.html),()  
=> replace(before.jpg,after.jpg)
```

これは、root.html に表示されている商用のパナー広告をクリックして index.html にアクセスしてきた利用者に対して、異なる画像を提供することができる例である。

また、利用者がページ index.html にアクセス要求をしたときに、サーバに蓄積されたルール実行履歴の中でルール名が Hanamaki であるようなルールを 50 回以上実行していたら、リンク hanamaki.html を追加する場合は、

```
index.html : (),(Hanamaki, times >= 50)  
=> add(hanamaki.html)
```

と記述する。この例は、利用者全体のアクセス履歴を用いてルールを記述する例である。

このようにページ作者は動的再構成ルールを記述することで利用者に対してコンテンツの異なったページを提供することができる。

3. メタルール

メタルールは動的再構成ルールに対して記述するルールである。メタルールは、動的再構成ルールの間の関係を記述したり、複数のページに対するルールに対して操作を行なったりすることができる。以下にメタルールの長所を示す。

- 動的再構成ルールへの操作

動的再構成ルールが記述されているファイルに対して操作をするルールを記述することができる。以下にメタルールの記述方法を示す。

```
rulefile_name : incidental,history => function
```

rulefile_name は動的再構成ルールを蓄積したルールファイル名である。そして、関数 function には、add, delete, replace の 3 つを用意した。add はルールを追加する関数、delete はルールを削除する関数、replece はルールの内容を変える関数である。例えば関数 delete は、日付が過ぎてしまったルールを削除するといった場合に用いる。図 1 にメタルールを用いた動的再構成ルール変更を示す。

- 動的再構成ルールの関係・禁則

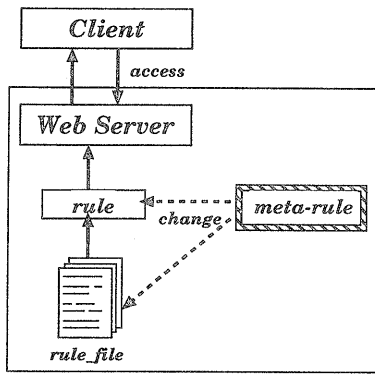


図1 メタルールによる動的再構成ルールの変更

メタルールは、動的再構成ルール間の関係や禁則を記述することができる。例えば、ページ作者が以下の3つの動的再構成ルールを記述したとする。

- 1 「7月から9月の間は夏模様の背景に置き換える」
- 2 「お盆の時期にはお盆用の背景に置き換える」
- 3 「お盆の時期にはお盆用の画像を追加する」

このような場合に、お盆の時期にはルール1とルール2を同時に実行することはできないが、ルール1とルール3やルール2とルール3は同時に実行することができる。メタルールでは、ルール1とルール2のようなルールの衝突が起こらないようにルール間の関係や禁則を記述することができる。メタルールによって動的再構成ルールの衝突の問題を解決してからルールを実行することができる。

- 動的再構成ルールを生成する変数付き再構成ルール

メタルールの属性に変数を記入しておき、利用者のアクセスから得ることのできる情報を変数に代入して動的再構成ルールに追加する。つまり、メタルールによって、新たに動的再構成ルールを生成することができる。例えば、利用者のアクセス場所ごとに見出しを変えたい場合に、場所を記述する代わりに変数を記述しておく。そして、利用者のアクセス時に得ることができるイベント情報から実位置の情報を変数に代入してページに表示する。

また、異なるページのアクセス履歴を比較し、その内容に基づいて動的再構成ルールを生成することができる。例えば、ページAとサイト内の他の全ページのアクセス数を比較し、ページAのアクセス数が最も多いときだけ他の全ページに対してルールを追加するといった場合に、ページ名に変数を用いることで全て

のページ名を記述することなく容易に動的再構成ルールを追加することができる。図2にページのアクセス数を比較しメタルールを用いて複数の動的再構成ルールを変更する例を示す。

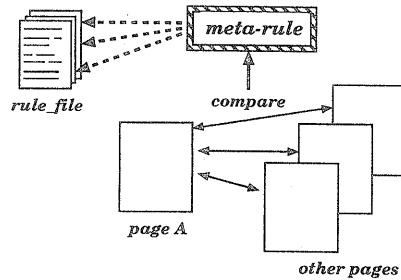


図2 変数を用いた複数の動的再構成ルールの変更

このように、メタルールは動的再構成ルールを操作するためのルールを記述することができる。メタルールを用いることで、より利用者ごとに異なったコンテンツを提供することができる。

4. ECA ルールとの比較

本研究で議論している動的再構成ルールとメタルールを用いたシステムをアクティブな Web システムと位置づけ、アクティブ Web システムとアクティブデータベースシステムの比較を行なう。アクティブデータベースは、イベント・コンディション・アクションの組を記述した ECA ルールによって様々なデータに対するアクセスを制御している。筆者らのアクティブ Web システムも ECA ルールを元にして、イベント・コンディション・アクションを用いて Web コンテンツのアクセス管理を行なっている。表1にこれをまとめる。

	Active DB	Active Web
Event	構造操作 トランザクション 例外	ページアクセス 時刻・位置・参照元 目標時間経過
Condition	データベースの 状態	利用者履歴 サーバの履歴情報 (ルール実行履歴)
Action	操作	ルールの実行 コンテンツ再構成 ページ活性化

動的再構成ルールとメタルールを用いたアクティブ Web システムと ECA ルールを用いたアクティブデータベースシステムとの違いとして、

- 利用者個別のアクセス履歴情報や利用者全体のアクセス履歴の集約情報をルールに用いている。
- ルールの衝突を解決してからアクションを実行する。
- メタルールにより新たにルールを生成することができる。

といったことが挙げられる。

ルール記述言語の仕様

動的再構成ルール (rule) とメタルール (meta-rule) を宣言的に記述するための言語仕様を以下に示す。

```

<rule> ::= <page_name> : <incidental> ,
          <history> => <function>
<page_name> ::= string
<incidental> ::= date | location | <referer>
<referer> ::= <URL>
<URL> ::= string
<history> ::= <user's history>
            | <server history>
<user's history> ::= [<times>] | [<valid_time>]
                  | [<old_info>]
<rule history> ::= (<rule ID> , <times>)
                  | (<combination> , <times>)

<times> ::= integer
<valid_time> ::= integer <span>
<span> ::= seconds | minutes | hours
         | days | weeks | months
<old_info> ::= (<rule ID> , <times>)
<rule ID> ::= string
<combination> ::= <rule ID> , <rule ID>
                | <rule ID> , <combination>
<function> ::= <function_name>
              (<resource>)
<function_name> ::= add | hide | replace
<resource> ::= <contents> | style
<contents> ::= <content>
              | <contents> , <content>
<content> ::= string

<meta-rule> ::= <rulefile_name> : <incidental> ,
                <history> => <meta-function>
<rulefile_name> ::= string
<meta-function> ::= <meta-func_name>
                  (<element>)
<meta-func_name> ::= add | delete | replace
<element> ::= <before_element> :
              <after_element>
<before_element> ::= string | integer
<after_element> ::= string | integer

```

5. プロトタイプシステム

プロトタイプシステムは、利用者のページアクセス履歴 (利用者履歴) とサーバに蓄積された全利用者に対して実行したルールの履歴 (ルール実行履歴) に基づいて、動的にページコンテンツの付加や隠蔽をして、ページ内容を変化させるシステムである。

利用者履歴の管理には、ページ利用者の情報を利用者のマシンに一時的に書き込み保存することができる仕組みである Cookie⁴⁾ を用いる。Cookie を用いることによって、利用者ごとに異なったサービスや広告を提供することができる。

本システムにおいて Cookie に蓄積する情報は、利用者履歴のみでイベント情報は蓄積しない。これは、利用者履歴のイベント情報はページアクセス時に決定するためである。

5.1 システムの構成

5.1.1 動的再構成ルールの定義

プログラミングなどの知識のないページ作者が容易にページ内容を変化できる動的再構成ルールを記述できるように、ルールへの入力には HTML の FORM タグを用いる。動的再構成ルールには、履歴情報とページコンテンツの依存関係を記述する。図3は利用者履歴に対する動的再構成ルールを入力する画面である。各フィールドで記入する内容は上から、

- ページを置くディレクトリの場所
- ルールを定義するページファイル名
- 時刻
- 場所
- 参照元 (どの URL から来たか)
- アクセス回数
- 前回のアクセスでのルール実行履歴
- 関数 と ページコンテンツ
- ルール名

である。イベント情報や利用者履歴のうちの必要な条件と関数やルール名などを入力する。

また、ルール実行履歴に対する動的再構成ルールは利用者履歴のときと同様に、

- ページを置くディレクトリの場所
- ルールを定義するページファイル名
- 利用者全体に対して実行したルールの名前あるいはルール名の組合せ
- ルール実行回数
- 関数 と ページコンテンツ
- ルール名

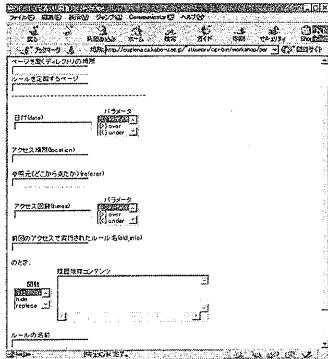


図3 利用者履歴に対する動的再構成ルールの入力画面

を入力する。

入力されたルールは CGI プログラムによって、ページごとに異なったファイルに蓄積される。利用者履歴に対するルールとルール実行履歴に対するルールもそれぞれ異なるファイルに蓄積される。

5.1.2 メタルールの定義

メタルールは動的再構成ルールと同様に HTML の FORM タグを用いて記述することができる。入力する内容も動的再構成ルールとほとんど変わらないが、関数は add, delete または replace を選択する。add を選択した場合は、利用者の履歴がルールの属性の条件を満たしたときに動的再構成ルールに新たにルールが追加される。delete を選択した場合は、イベント情報やアクセス履歴がルールの属性の条件を満たした場合に、ルール名が一致する動的再構成ルールを削除する。replace を選択した場合は、条件を満たした場合に動的再構成ルールの該当する文字列を全て別の文字列に置き換える。

5.1.3 全体構成

利用者がページにアクセスすると CGI プログラムは次の処理を行なう。

1. 利用者のページアクセス要求に含まれるイベント情報と Cookie に蓄積された利用者履歴を読み取る。
2. 各ページごとに蓄積されたルール実行履歴を読み取る。
3. 利用者個別と全体に対して定義した動的再構成ルールとメタルールを読み取る。
4. 1から3のデータを比較して、利用者に対して提供するルールを仮決定し、メタルールによってル

ル間の関係の問題を解決する。場合によってはルールの削除や置換を行ない、問題が解決されたルールの内容を本決定とする。

5. 本決定されたルール名を含めた利用者履歴を Cookie に蓄積する。同時に実行したルール名の組合せをルール実行履歴に蓄積する。
6. 本決定されたルールの内容に基づいたページコンテンツが利用者提供される。

この処理は、利用者がページにアクセスする度に行なわれる。プロトタイプシステムの全体構成を図4に示す。

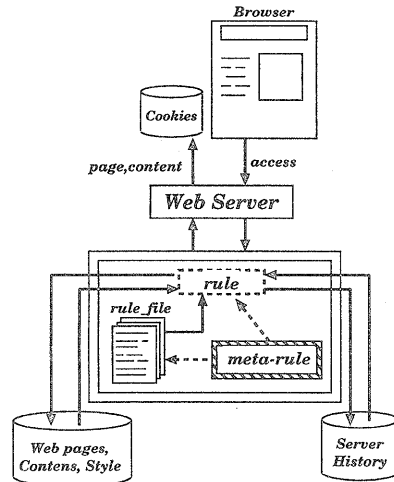


図4 宣言的ルールによるページの動的再構成

5.2 実行例

本システムによって定義する動的再構成ルールとメタルールを用いた Web ページの実行例を示す。以下は、オンラインショッピングのページ `index.html` に利用者がアクセスした際、履歴によってページ内容が変化される例である。動的再構成ルールとメタルールは次のように定義した。

```
[1]index.html : (),(times == 1)
                => add(「はじめて」)
```

`index.html` へはじめてのアクセスの場合には「はじめて」を付加する。

```
[2]index.html : (),(times > 3)
                => replace(1万,7千5百)
```

通算4回以上のアクセスであったら値段を1万から7千5百に置き換える。

```
[3]index.html : (),(change , times > 9)
=> replace(notePC,coffee_cup)
```

利用者全体に対してルール名が change であるようなルールを通算10回以上実行したらノート PC 関連のコンテンツを全てコーヒーカップに変化する。

まず、利用者がはじめて index.html に訪れた場合、ルール1の属性の条件を満たすのでルールが実行されページにコンテンツが付加されるが、他のルールは条件を満たさないため実行されない。図5にこれを示す。



図5 ページ内容の動的再構成の例1

通算4回以上で再び index.html にアクセスした場合は、ルール2の条件を満たすため、机の値段が置き換わる。また、すでに1回目のアクセスではないため「はじめに」は表示されない。図6にこれを示す。

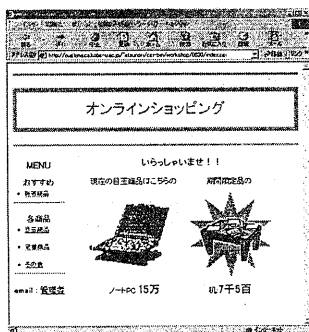


図6 ページ内容の動的再構成の例2

また、ページ index.html でルール名が change であるルールを利用者全体に対して10回以上実行している場合には、ページに表示するコンテンツのうちノート PC をコーヒーカップに置換している。このルールはルール実行履歴の属性が条件を満たすれば、初めて訪れる利用者にも実行される。図7に示す。



図7 ページ内容の動的再構成の例3

さらに、本システムを用いた別の例を挙げる。以下の例は、利用者の位置情報を取得することによって、表示するコンテンツを変化させる岩手県の観光案内ページの例である。最初のアクセスでは、表示されるコンテンツは宮澤賢治記念館の写真が表示されている(図8)が、温泉街に移動しそこでこのページにアクセスすると温泉街案内の地図が表示されている(図9)。



図8 地理情報を用いた例1

この実行例では、利用者から位置情報を入力してもらうことで情報を得ている。しかし、将来的には利用者の実位置を自動的に獲得できるようなシステムを想定している。これには、クライアントから地理情報システムに自分のいる場所を問い合わせるといった操



図9 地理情報を用いた例2

作を行なうプラグインが必要となる。一般的なサーバがクライアントに対して発行する Cookie 情報は異なっているため、我々はこのようにして獲得できる地理情報を Geo-Cookie と呼んでいる。本システムと Geo-Cookie を用いることで、図8や図9といった変化例を自動的に利用者に対して提供することができる。

6. まとめ

本稿では、Web ページをコンテンツ単位でアクセス権管理するために、ページコンテンツの表示にルールを設け、利用者のアクセス時のイベント情報や利用者個人の閲覧履歴、全体のアクセス履歴に基づいてページ内容を変化させることを提案した。そして、プログラミングの知識を持たないページ作者でも容易に動的に異なるコンテンツを提供するためのツールとして動的再構成ルールと、そのルールを管理するメタルールについて述べた。実装したプロトタイプシステムでは、HTML の FORM タグを用いることで容易にルールを記述することができ、実際にイベント情報や2種類の履歴に基づいて同一ページ内で異なるコンテンツを提供した。そして、メタルールによって動的再構成ルールの操作を行なうことができた。

プロトタイプシステムでは、イベント情報として利用者のアクセス時刻や位置などをルールに記述することができるが、利用者によっては現在の時刻や位置よりも未来のある時間やある場所といった仮想的な時刻や位置の情報の方が有用である場合がある。例えば、利用者が交通機関で移動中にページにアクセスした場合には、利用者にとっては現在移動中の場所の情報よりも数時間後の目的地の情報の方が有用である場合が多い。プロトタイプシステムではこのような場合にも、利用者が仮想時刻・仮想位置の情報を持ってアクセスを行なうことで、その利用者に対して仮想的な情報を

提供するという応用が可能である。

筆者らは、利用者の閲覧履歴やサーバに蓄積されたアクセス履歴情報および、このような時間と空間の情報をを用いてページコンテンツのアクセス権管理を行ない、利用者ごとにパーソナライズされたページを提供することを目的としている。今後の課題としては、

- 時間・空間情報を考慮した動的再構成ルールやメタルールの記述力の強化
- XML(eXtensible Markup Language)^{5),6)}を用いたプロトタイプシステムの応用
- プロキシ経由のアクセスに対するシステムの改善

などが挙げられる。特に、プロキシサーバにはキャッシング機能があり、クライアントからプロキシを経由してページにアクセス要求をしようとした場合、プロキシサーバのキャッシュに蓄えられた内容をプロキシサーバで返してしまい、実際に目的のページにアクセスしたことはない。そのためアクセス履歴によるページの内容の変化が行なえないといった問題点があり、プロキシ経由に対するシステムの改善が重要であると考えている。

謝 辞

本研究の一部は、文部省科学研究費基盤(C)「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)および、日本学術振興会未来開拓学術研究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)によっております。ここに記して謝意を表すものとします。

参 考 文 献

- 1) 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. 第11回データ工学ワークショップ論文集, 2000.
- 2) 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.44, pp.17-24, 2000.
- 3) Norman W.Patom. "Active Database System". *ACM Computing Surveys* Vol.31, No.1, pp.63-103, 1999.
- 4) Client Side State - HTTP Cookies.
<http://home.netscape.com/newsref/std/cookie.spec.html>.
- 5) W3C - The World Wide Web Consortium.
<http://www.w3.org/>.
- 6) Japan XML User Group Web Page.
<http://www.xml.gr.jp/>.