シングルページのトラッキング

隅谷 孝洋, 稲垣知宏, 長登康, 中村純

広島大学 情報メディア教育研究センター 〒 739-8521 東広島市 鏡山 1-7-1
E-mail: {sumi,inagaki,nagato,nakamura}@riise.hiroshima-u.ac.jp

あらまし 複数ページからなる階層構造を持ったコンテンツを WebCT で提示する場合、シング ルページツールを利用すると簡便であるが、トラッキングツールが利用できないと いう欠点がある。ここでは、アクセスログを解析することにより、WebCT トラッキ ングツールと同程度またはそれ以上の機能を実現する事を試みる。また、教師が特に アクセスログを解析するという意識を持たずに自然にこの機能が利用できるようなイ ンターフェースについて議論する。

キーワード シングルページ, ページトラッキング, 学生トラッキング, アクセスログ解析

Tracking on Single Page

Takahiro SUMIYA, Tomohiro INAGAKI,
Yasushi NAGATO and Atsushi NAKAMURA

Information Media Center, Hiroshima University
1-7-1, Kagamiyama, Higashi-Hiroshima, 739-8521, JAPAN
E-mail: {sumi,inagaki,nagato,nakamura}@riise.hiroshima-u.ac.jp

Abstract Using Single File page tool, we can conveniently supply a learning object which consists of multiple files, rather than using Content Module tool. In this case, the tracking tools on WebCT are not available. Here, we report a method to tracking the files those are supplied from Single File page tool, by analyzing httpd access log directly.

Keywords Single Page, Page Tracking, Student Tracking, Access log file analysis

1 はじめに

Web を通した学習の場を提供する際に、単なる Web ページを開設するのみではなくてコース管理ツールを使う理由のひとつとして、学習履歴の把握が容易なことが挙げられる。WebCT にももちろん「ページトラッキング」「学生トラッキング」というかたちで学習履歴を調べる機能が登載されている。

しかし、これらのトラッキング機能はそれ程高機能ではないうえに、コンテンツモジュールとして提示されるコンテンツファイルしかトラッキングの対象にならない。

コンテンツモジュールをトラッキングする際に、機能を補完してより詳しい解析を行なう試みが井上ら [1], 山川ら [2][3] などによってなされている。井上らは WebCT のページトラッキング機能と httpd のアクセスログを組み合わせて詳細な解析を行なっている。山川らは、WebCT がデータベース領域に保存するログファイルを直接解析するための枠組みを提供している。

ここでは、WebCT が保存するログファイルは参照せず、www サーバのログファイルを読み取ることにより、コンテンツモジュールとシングルページの両ツールで配置されたファイルのトラッキングを行なうことを試みる。

われわれの立場は、WebCT の利用者であると同時

に管理者でもあり、学内の WebCT 利用教官によりよい利用環境を提供する必要がある。この報告 (と今後の開発) の目的は下記である。

- WebCT 単体ではまったく提供されない、シングルページのトラッキング情報を得ること。
- 上記情報とコンテンツモジュールのトラッキング情報と併せて、デザイナが手軽に参照できるような環境を構築すること。
- トラッキング情報を通して活動状況の可視化や何らかの意味での指標化などを行なうこと。

2 httpd のログを元にしたトラッキング

文献 [2] に詳しく示されているように、WebCT はコンテンツモジュールの各目次項目へのアクセスのタイミングを記録している。基本的にはこの記録を用いて、WebCT はトラッキング情報を提供しているのだが、これを利用しようとするといくつかの問題があった。

まず、この報告の主目的のひとつでもあるのだが、シングルページへのアクセスが記録されていない。

次に、限られたイベントしか記録されていないことも問題である。たとえばページ A を 5 分参照後コースホームページへ移動、5 分後ページ B に移動したとする。ページ A の参照時間は 5 分となるべきだが、コースホームページへの移動というイベントは記録されないため、参照時間は 10 分となってしまう。

さらに、アクセス元 (リモートホスト) も記録されていない。またこれは表示上の問題に過ぎないとも言えるが、ページタイトルのみの表示でありコンテンツモジュールの構造が考慮されていない。

以上の理由により、httpd のログからトラッキング情報を構成できないかと考えた。またそこから得られた知見を、WebCT 以外の一般的なアクセスログ解析に利用することも期待できる。

2.1 ログフォーマット

WebCT 4.0CE で利用されている httpd (HTTP Daemon, WWW サーバプログラム) は apache2.0.x である。クライアントである WWW ブラウザから apache へのリクエストは逐一ログファイル

```
${webctdir}/server/logs/access_log
```

に記録されていく。ここで `${webctdir}` は WebCT のインストール時に指定したディレクトリパスに `webct/` を加えたものである。このファイルには一行にひとつのリクエストが記録されているが、そのフォーマットは apache の設定ファイル

```
%h %l %u %t \"%r\" %>s %b %T
```

%h リモートホスト
%l リモートログイン名 (identd から取得する)
%u リモートユーザ名 (apache 認証によるもの)
%t 時刻
%r リクエストの最初の行
%>s 終了ステータス
%b HTTPヘッダを除いた送信バイト数
%T 処理時間（秒）

図 1: アクセスログ内の項目の並び

```
${webctdir}/server/conf/httpd.conf
```

の `LogFormat` と `CustomLog` ディレクティブで規定される。デフォルトでは図 1 のような並びで各項目の情報が記録される。

通常 WWW ブラウザに表示されている一画面は HTML ファイル、画像ファイルなど複数のリソースからなっているため、一画面分のリクエストを受けつけるとログファイル上は数行の記録がなされることになる。

2.2 コースとユーザの特定

通常の Web サイトであれば、アクセスログに記されたパス (図 1 の `%r` の部分から読み取れる) を見れば、どのリソースへのアクセスであるかは直ちにわかる。アクセスログのパスは、httpd.conf の `DocumentRoot` ディレクティブで指定されたディレクトリからの相対パスである。なので、たとえば `DocumentRoot` が `/home/apache/docs` の時に、アクセスログにあるパスが `/foo/hoge.html` であればその行は、ユーザのリクエストが最終的にはサーバ上のファイル

```
/home/apache/docs/foo/hoge.html
```

になることを示している。

WebCT サーバの場合、もう少しややこしいが、httpd.conf の

```
DocumentRoot
ScriptAliasMatch
ScriptAlias
```

の記述を読み、ディレクトリ内容を調べてみると、アクセスログ内のパスに関して概ね以下のようなことが言えるようである。以下で `${course}` はコース ID に置き換わる部分である。

(1) `/${course}/...`
コースファイル (myFiles へ置いたもの) へのアクセス

(2) `/web-ct/...`
WebCT ツールが使う部品 (HTML コードやイメージなど) へのアクセス

(3) `/web-ct/course/${course}/...`
コースに配置されたツールの利用によってシステムが作成したファイルへのアクセス。バックアップファイル、コンテンツモジュールに配置されたファイルのコピーなど。

(4) `/wct_files/...`
「ファイル管理」で MyFiles の下に出てくる「WebCT-Files」内のファイルへのアクセス。

(5) `/webct/...`
コースの外 (myWebCT, エントリーページ、管理ページなどなど) で使用する様々なツールへのアクセス。

(6) `/SCRIPT/${course}/...`
コースに配置されたツールへのアクセス。

なので、(1),(3),(6) についてはパス内のしかるべき場所を見れば、どのコースへのアクセスであるかは判断できる。それ以外のパターンでは判断不能であると思われるが、今回の目的 (コンテンツファイルへのアクセスを調査する) にはさしつかえない。

また、(1),(3),(6) に相当するディレクトリには全て .htaccess ファイルが作成されており、アクセスログにリモートユーザ名が記録されるようになっている。ここに記録されるリモートユーザ名は WebCT にログインするときの ID と一致する。すなわち、アクセスログの各行に注目した時に、それがどのユーザのアクセスで、どのコースを利用しているのかと言う事が容易に判断できる。

逆に特定のユーザのアクセス、特定のコースへのアクセスを抽出することも容易である。たとえばユーザ foo のアクセスやコース bar へのアクセスを抽出するための UNIX コマンドの例を挙げると

```
% awk '$3=="foo"{print}' access_log
% egrep \
   '(GET /bar/|(GET|POST) /SCRIPT/bar/)' \
    access_log
```

などとなる。ここで抽出されるのは foo の (1),(3),(6) のパターンのアクセスとコース bar への (1),(6) のパターンのアクセスだけであることに注意が必要である。各コース内のページ／ツールへのアクセスに伴って (2),(4) のパターンが多く発生するのだが、これらは上記では抽出されない。

2.3 どの行がコンテンツファイルへのアクセスなのか

シングルページ シングルページの場合は単純である。myFiles の中に `foo.html` を置き、それをコースホームページにシングルページとして配置したとする。学生がそのシングルページにアクセスすると、ログファイルには (1) のパターンの記録が残る。すなわち

```
GET /${course}/foo.html
```

となる。

上のような行をカウントすると、シングルページへのアクセス回数をカウントしていることになるだろうか？答は、状況により Yes である。`foo.html` へのリンクが含まれているファイル (`bar.html` とする) があるとして、それをコンテンツモジュールに配置したとしよう。学生がコンテンツモジュールの項目 `bar.html` を閲覧した場合、後述するように (1) の形式ではログに残らない。しかし、`bar.html` 内のリンクをクリックして `foo.html` を表示させたときには上のような (1) の形式でのログが残るのである。

コンテンツモジュールの場合 WebCT の提供するツールへアクセスした場合、主に (6) の形式のログが残る。たとえばコースホームページを開けば

```
GET /SCRIPT/${course}/scripts/
                /student/serve_home?...
```

最後省略しているが、`serve_home` という CGI プログラムが引数とともにリクエストされる。掲示板であれば `serve_bulletin`、シラバスツールであれば `serve_syllabus.pl` と言うように、ツールごとに異なる CGI プログラムが呼び出される。

コンテンツモジュールを表示するときに使われる CGI プログラムは、`serve_page.pl` と言うファイルのように見える。ログを見ると引数にファイル名らしきものも確認でき、これを調べれば良いように最初は思えた。ところが、ページを表示させても `serve_page.pl` が呼び出されない場合もある。おそらくキャッシュの関係だろうと思うのだが、よくわからない。

で結局どうしているかと言うと、`button_bar` という CGI プログラムに注目している。このプログラムは、コンテンツモジュール内のナビゲーションを生成しているようだ。このプログラムが

```
button_bar?1137703098+672717578
```

と言うような形式で呼び出されたときと、利用者がコンテンツモジュールのページを開いたときは一

致しそうである[1]。二つの数値の引数があるが、一つめはコンテンツモジュール内の目次項目に付けられた内部的な ID[2]、二つ目はコンテンツモジュール自体に付けられた ID である。

ID 1137703098 の項目が具体的にどのファイルに対応するかは、

```
${webctdir}/webct/courses/${course}
        /database/paths/672717578.pth
```

を見るとわかる。また、目次項目に記されているタイトル文字列は

```
${webctdir}/webct/courses/${course}
        /database/pages/__WCT__pages.pgs
```

にある。

2.4 情報の抽出と集計、提示

あるコンテンツファイルを特定したときに、それがシングルページとして提示されていてもコンテンツモジュールとして提示されていても、前節に記したようにログファイルを読めば誰が何時どこからアクセスしたかが判断できる。

ここでは、WebCT のページトラッキング機能が提供しているように、コースコンテンツの一覧と、それ等に対するアクセス情報を提供することを考える。

そのためには、コースを特定したときに、そのコースに配置されているコンテンツモジュールとその目次項目、シングルページの一覧を入手する必要がある。この情報は、

```
${webctdir}/webct/courses/${courses}
            /database/paths/_homepage.idx
```

から順次辿って行く事により得られる。上のファイルにはコースホームページに配置されたページ/ツールの一覧がテキスト形式でおさめられている。シングルページの場合は、記された ID を`${sid}`とすると

```
${webctdir}/webct/courses/${courses}
              /database/pages/${sid}.att
```

なるファイルの中に対応するファイル名がある。コンテンツモジュールの場合は前節に記した方法で辿ることができる。(図 2 参照)

上記方法でコース内のコンテンツ一覧を調べ、対応するリクエストをアクセスログから抽出するプログラムを作成した。出力結果を図 3,4 に示す。

図 3 はコンテンツモジュール中心のコースの場合の出力である。WebCT 内蔵のトラッキングツールの場合、目次の並びや階層構造がまったくわからないと言う問題もあるが、今回作成したページでは、図に示したように元の階層構造をそのまま活かして表示している。図に示しているのはアクセス回数だけであるが、材料は全て収集ずみなので、曜日毎のアクセス数推移であるとか、個人毎のトラッキング等にも容易に拡張可能である。

図 4 はシングルページ中心のコースの場合、INFOSS 情報倫理テキストが掲載してあるコースである。INFOSS 情報倫理テキストのように、複数の HTML ファイルが階層構造を構成しており、内部で完結したナビゲーション機能まで持っている場合、トップページに相当する部分だけをシングルページとして指定してやれば、学生に全体を提示することができる。この場合、前述の方法で検索しても、コースのコンテンツファイルとしてはトップページしか出てこない。トップページからリンクがはられているページは、コースの定義には登場しないけれども、アクセスログには記録が残ることになる。図の「Unknown File(s)」となっているところには、そのようなファイルの一覧が表示されている。

上記のような状態は、シングルページだけでなくコンテンツモジュールでも起こり得る。典型的な例が、Microsoft Powerpoint のスライドショウを HTML 変換したものをアップロードした場合である。通常はスライドショウの起点となるページをコンテンツモジュールの目次項目として設定するが、二枚目以降のファイルの登録は行なわないからである。この場合二枚目以降へのアクセスは WebCT のトラッキング機能では記録されない。

こうして作成されたページを、どうやってデザイナに提供するかと言う事であるが、この目的には My-Files が利用できる。定期的に、たとえば日に一回とか週に一回このページを生成するようにし、その結果を各デザイナの My-Files 領域に保存しておけば、デザイナはファイル管理ツールのプレビュー機能を使って何時でも見られるし、またデザイナ以外には見られないようになる。

3 おわりに

今回ページトラッキング、学生トラッキングに必要なデータを収集する方法が、若干あやしいにしても確立できたので、今後はさらにいろいろな角度からデータを眺められるようにページを拡張をしていく方針である。

学生トラッキングの重要な要素のひとつである、「ページの閲覧時間」の算出は非常に難しい。WebCT はここは結構いいかげんで、ページ閲覧、クイズ、掲示版利用など非常に限られたイベントの発生時間だけを記録し、閲覧時間を算出している。

[1] これはスクリプトファイルを精査して出した結論と言うわけではない。

[2] 3.8CE まではこの位置にファイル名がきていた。

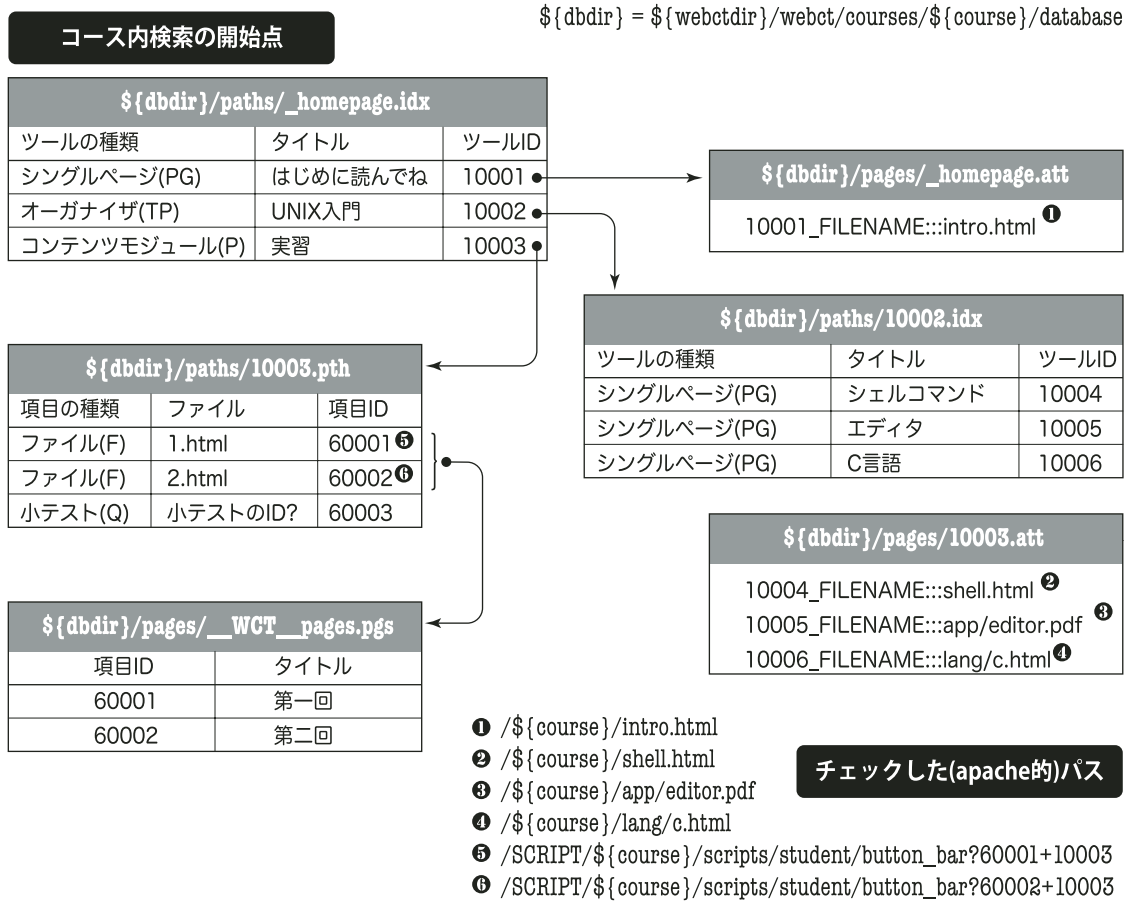

図 2: コース内に配置されたコンテンツファイルの検索と、対応するパスの設定

もう少しマシな実装はあるだろうが、おそらく一番問題になるのは、最後に表示したコンテンツページの閲覧時間を測る合理的な手段が存在しないことだろう。コンテンツページを参考文献的に使っている場合、最後に表示したページが学生にとってもっとも重要なページである事が多いだろう。

これを克服するため、発想を転換して、ブラウザ側から情報を送ってもらうようにすることはできないだろうか。HTML の META タグで一定時間後の再ロードは指定できるし、おそらく JavaScript でも同様の仕掛は実現できるのではないだろうか。再ロードがかかればサーバ側に記録が残るわけなので、少なくともブラウザ画面に表示されつづけている時間を測ることが可能になる。CMS で HTML ファイルを送出する際にこうした仕掛をしてしまうわけである。結構良いアイデアだと思うのだがどうだろうか。

本報告では一般的に不要と思われる細部まで言及したつもりだが、紙幅の都合で中途半端な記述になっているところも多々ある。不明の点は著者に問い合わせて頂きたい。本報告が、同様の部分を hack しようとするユーザの一助となれば幸いである。

参考文献

[1] 井上 仁, 多川孝央 “履歴情報に基づく講義の分析” 第 1 回 WebCT 研究会,2003

[2] 山川 修, 田中武之, 菊沢正裕 “LMS を使った学習プロセスの分析と評価” 第 1 回 WebCT 研究会,2003

[3] 山川 修, 田中 武之, 菊沢 正裕 “学習履歴情報の詳細分析のための枠組み” 第 2 回日本 WebCT ユーザカンファレンス,2004

[4] 隅谷 孝洋, 稲垣 知宏, 長登 康, 中村 純 “WebCT のカスタマイズ” 第 1 回 WebCT 研究会, 2003

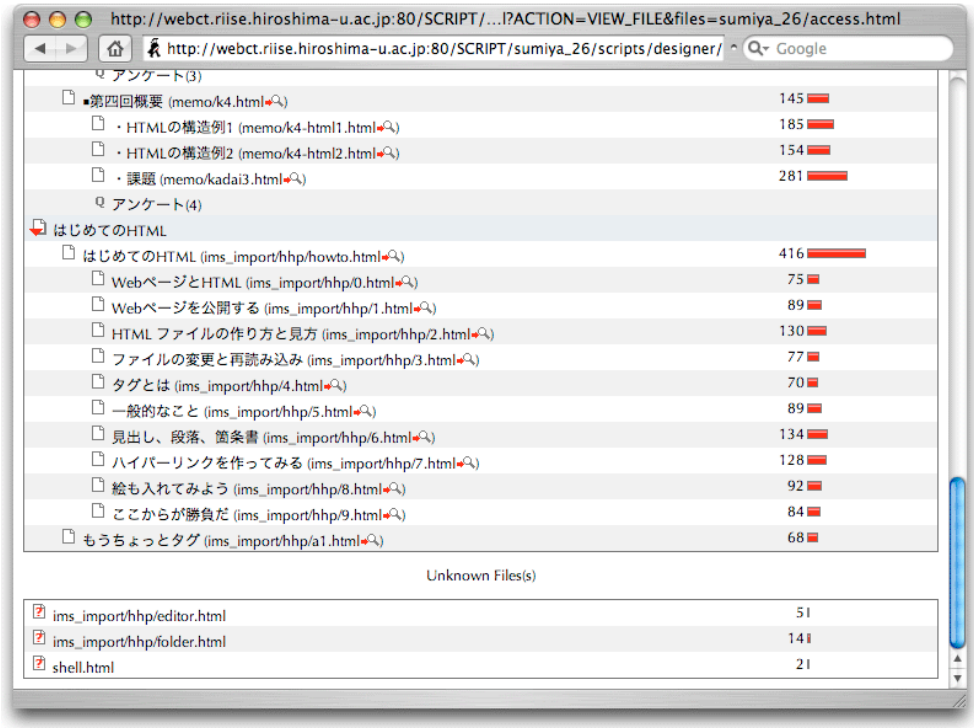

図 3: アクセスログ集計結果 (1)

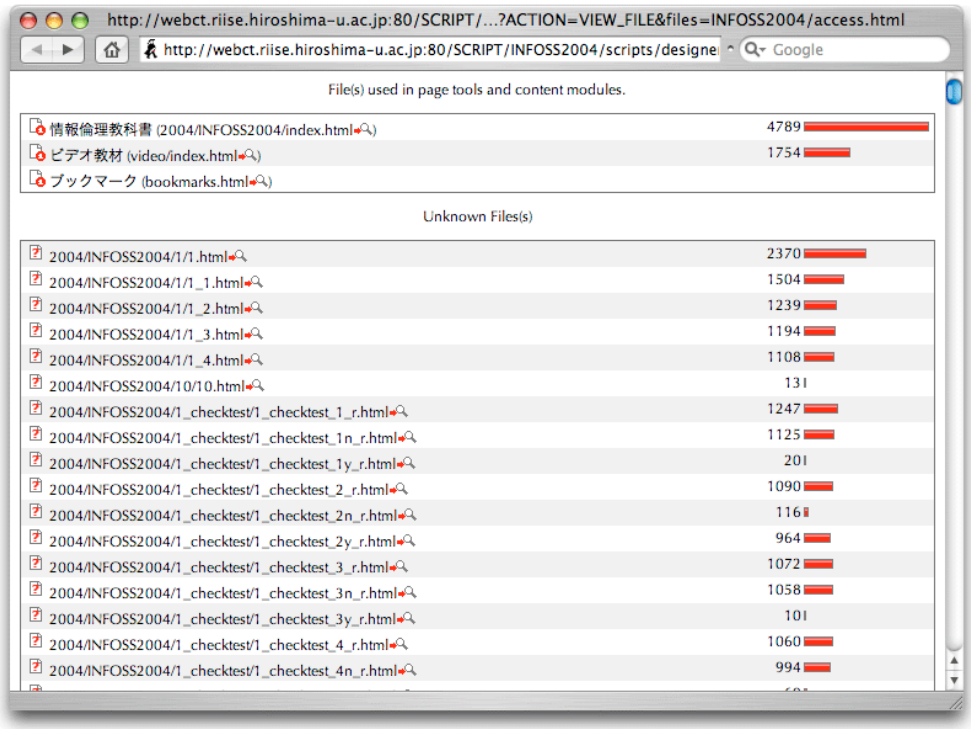

図 4: アクセスログ集計結果 (2)