

WebCT のカスタマイズ

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@riise,inagaki,nagato@riise,nakamura@riise}.hiroshima-u.ac.jp

あらまし: WebCT はシステムの大部分が Perl で書かれており、その気になればかなりのカスタマイズが可能である。ここでは、プラグイン機能を使うまでもないような小規模な機能追加や、サイトごとのカスタマイズについて、その手法と広島大学の試験用 WebCT サーバに適用してみた結果を報告する。

キーワード: WebCT, カスタマイズ, WebCT サーバ管理

Customizing WebCT

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@riise,inagaki@,nagato@riise,nakamura@riise}.hiroshima-u.ac.jp

Abstract: As the large part of WebCT system is written in Perl, we can make various customization by ourselves. We have investigated some small supplements of new features to WebCT system. In this paper, we report the technique for customization and results on our proof WebCT server.

Keyword: WebCT, Customize, WebCT Server Management

1 はじめに

筆者らのグループは大学内の情報インフラ整備の一環として WebCT システムを導入・運用し全学にサービスを提供している。WebCT は大変優れたコース管理システムであるが、導入サイトでの運用方針にそぐわない部分や細かな不具合などは学内での普及の妨げとなるため、可能な限り排除していかなくてはならないと思っている。

[1] において、本学の計算機システムのアカウント体系・講義登録システムに合わせた学生管理機能を、WebCT 外部の CGI として実現する方法について紹介した。このように外部 CGI を用いたり、もしくはプラグイン機能を用いたりして、WebCT に新たな機能を追加することが可能であるが、細かな修正や運用方針に合わない仕様を変更することはできない。

幸い、WebCT のシステムの大部分はスクリプト言語の Perl を用いて書かれており、この部分については導入組織においてある程度の修正やカスタマイズが可能である。Perl+CGI という仕組みは Servlet や JSP といった技術と比べるとプリミティブで見劣りするかもしれない。しかしカスタマイズする立場からすると、ソースが見えているのはもちろん、変更が即座に反映されるのだから非常にやりやすい。

とは言っても、変更箇所無くしては運用がままなら無い程の大きな変更を加えてしまえば、不具合が発生したときに対処が難しくなる。変更するにしてもすぐに元の状態に戻せる程度の些少なものととどめておくべきだろう。弱腰だけれどシステム全体のコードを把握することなど無理なのだから、サポートを受ける立場としては仕方がない。

ここでは、いくつかの実例を通し、Perl を修正しての WebCT をカスタマイズについてその手法と本学の試験用 WebCT サーバに適用してみた結果を報告する。以下は全て Linux サーバで確認した事項であり、その他のプラットフォームでは異なる場所があるかもしれない。

2 WebCT システムのファイル構成

カスタマイズの事例に入る前に WebCT のディレクトリ構成を確認しておこう。WebCT を構成するファイルは非常にたくさんあるが、大まかなディレクトリ構成は図 1 のようになる。最上位のディレクトリ `webctdir` は WebCT をイ

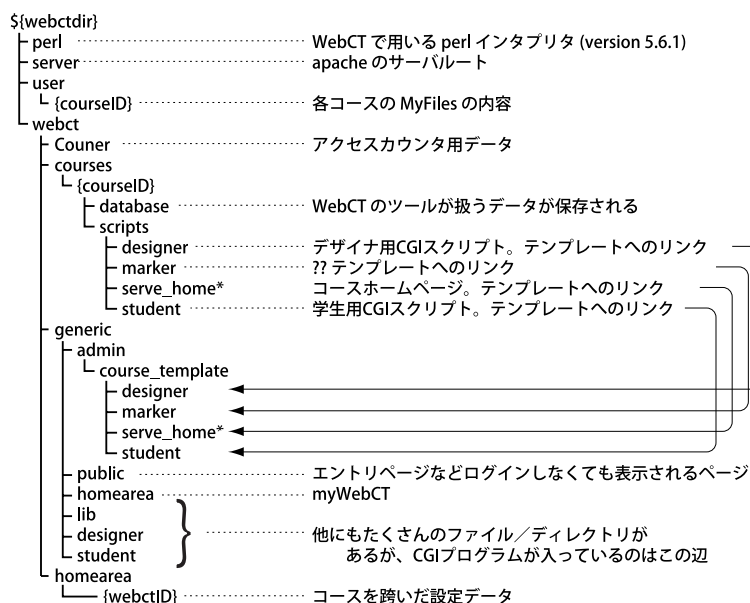


図 1: WebCT のディレクトリ構成

インストールしたときに指定したディレクトリ中の webct ディレクトリを指している。

実際の CGI のスクリプトが格納されているのは `${webctdir}/webct/generic/` である。一方、httpd サーバソフトウェアである apache に設定されている DocumentRoot は `${webctdir}/user/` であり、スクリプトが入っているディレクトリとは別のディレクトリツリーとなっている。WebCT では apache の ScriptAlias 機能と JavaScript を利用してユーザのリクエストを `${webctdir}/webct/generic/` 以下のスクリプトへと導いている。2,3 の実例を紹介しよう。

まず、WebCT のエントリーページを見てみよう。http://webct.hoge.jp/ のような URL をリクエストすると、javascript による URL の書換えを経て最終的には

```
http://webct.hoge.jp/webct/public/home.pl?action=print_home
```

という URL でエントリーページが表示される。Apache の設定ファイル¹ には

```
ScriptAlias /webct/public/home.pl ${webctdir}/webct/generic/public/home.pl
```

という記述があるので、上記 URL をリクエストしたときに実行される CGI スクリプトは

```
${webctdir}/webct/generic/public/home.pl
```

だということになる。

次に、コースのホームページを見てみよう。コース ID が foo であるコースのホームページを表示させると、URL は

```
http://webct.hoge.jp/SCRIPT/foo/scripts/serve_home
```

となっている。これの実体はどのファイルだろうか？再び Apache の設定ファイルを見ると

```
ScriptAliasMatch ^/SCRIPT/([A-Za-z0-9._?~]+)/scripts/(.*)
    ${webctdir}/webct/courses/$1/scripts/$2
```

という一行がある。上述の URL はこのパターンに適合し、実際に参照されるファイルは

```
${webctdir}/webct/courses/foo/scripts/serve_home
```

であることがわかる。このファイルを確認すると

¹`${webctdir}/server/conf/httpd.conf`



図 2: エントリーページ近辺のカスタマイズ結果

```

${webctdir}/webct/generic/admin/course_template/scripts/serve_home

```

へのシンボリックリンクになっており、最終的に実行されるのは上記のファイルであることがわかる。

後者の例で見たように、コースのホームページ以下で参照されるスクリプトは基本的に全て

```

${webctdir}/webct/courses/{コース ID}/

```

以下のスクリプトであるが、これは実は全て `${webctdir}/webct/generic` 以下のファイルへのリンクになっている。なぜこのような参照の仕方になっているかというと、コース毎のアクセス制限を `.htaccess` ファイルで行っているからである。おそらく初期の設計を未だに引きずって来ており、このような意味泥臭い実装になっているのだろう。

一方このようなプリミティブな実装であるがゆえに、カスタマイズする側にとっては都合がよい面もあるのではあるが、この辺りは将来のバージョンでは改善されて行くところだと思う。

3 カスタマイズの事例

3.1 エントリーページ近辺のカスタマイズ

利用者が最初に WebCT サーバにアクセスした際に表示されるエントリーページは、貴重な広告スペースである。サーバのメンテナンス情報や設定変更の広報等は myWebCT にも表示できるが、こちらのほうが遥かに目立つ。また、ログインする前に表示される部分なので、「どうやってログインするのか?」といった情報を最も手早く表示するスペースとしても貴重である。この部分のカスタマイズについては管理者マニュアルにも記述があるので、実際に行っている組織も多いだろう。

2 で見たように、エントリーページは `${webctdir}/webct/generic/public/home.pl` に引数を与えて作成されている。スクリプトを読んでみると、

```

${webctdir}/webct/generic/public/template/ja_template.html

```

と言うファイルを雛型として HTML コードを作成 (デフォルト言語が日本語の場合) しているので、これを編集すればよい。最上部の “WebCT へようこそ” と書かれた部分を “広島大学 WebCT サーバへようこそ” と変更した。さらに、お知らせのため最下部の `</body>` の前に

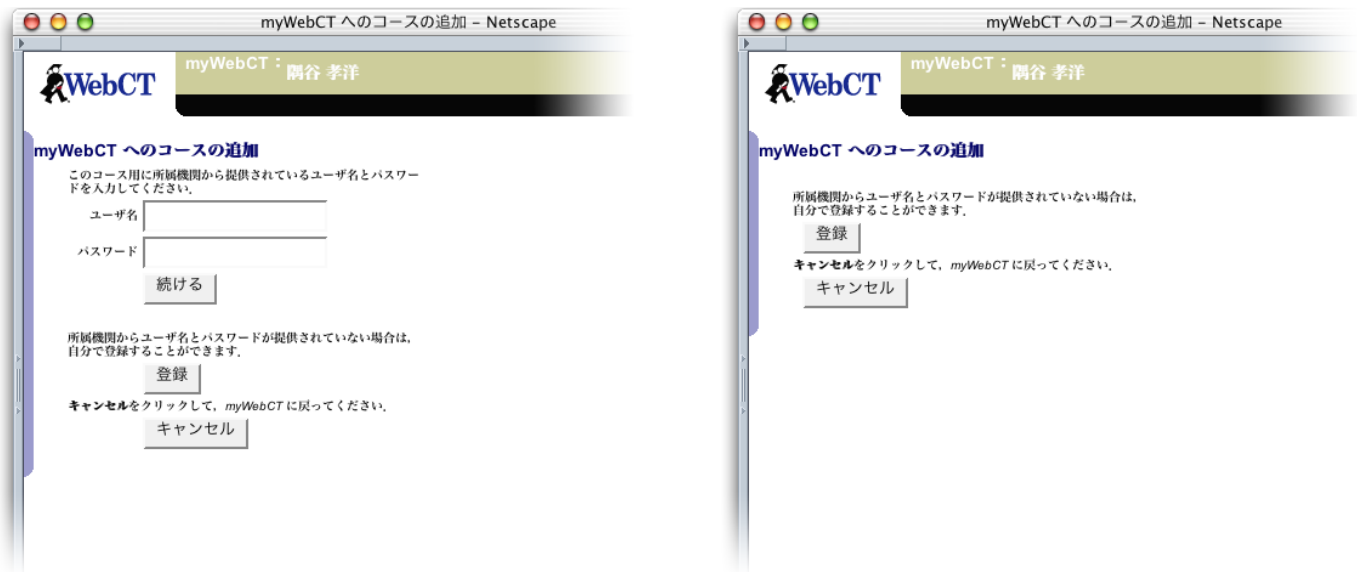


図 3: 不要な表示の抑制

```
<div style="border: 1px solid #CCCCCC; background-color: #FFFFCC;
margin: 0px 50px 0px 50px; padding: 10px 10px 10px 10px; width: 350px">
  <p><span style="font-size: larger; color: #CC0000">サーバ停止のお知らせ</span><p>
  <p>サーバリプレイスのため 8/15~16 の予定で WebCT サービスを停止します。</p>
</div>
```

を加えた。これでエントリーページを表示してみると図 2 左のようになる。

管理者マニュアルを参照すると、ユーザ名とパスワードを投入するログインページのカスタマイズについても記述がある。こちらは

```
`${webctdir}/webct/generic/ticket/ja_lgin.html
```

を改変すればよい。例えばログインに失敗したときに、エラーメッセージと共にパスワードの入手方法などを表示すると有用だろう。60 行目にエラーメッセージを作成している部分があるので、ここを書換えて例えば図 2 右のような表示にできる。

3.2 不要な表示の抑制

学生が myWebCT にコースをセルフ登録する際には図 3 左のような画面が表示される。セルフ登録を許可されているコースの場合は「登録」ボタンをクリックするだけでいいのだが、その上にユーザ名とパスワードを記入する欄があるため混乱する学生が非常に多かった。本学ではこの欄に記入できる情報の利用を許可していないため、全く不要な表示である²。

そこでこの欄を表示できないようにできないかと考えた。このページの URL は

```
http://webct.hoge.jp/webct/homearea/mywebct_manager.pl?ADD_COURSE_DO_IT+{コース ID}
```

となっている。Apache の ScriptAlias により、mywebct_manager.pl の実体は

```
`${webctdir}/webct/generic/homearea/mywebct_manager.pl
```

²そのような設定（「グローバルデータベース中にレコードがない学生を追加」を許可しない）にしているにも関わらずこの欄の表示が抑制されないと言うことは何か他の使い道があるのかも知れない。ご存知の方は教えてください

である。このスクリプトの中の `mywebct_add_course_account()` という関数呼出で HTML コードが生成されているようだが、この関数の定義は `mywebct_manager.pl` の中には無い。スクリプト冒頭で数個の他のスクリプトを `require` しているので、その中のどれかにあるはずである。適当にあたりをつけて³、

```
{webctdir}/webct/generic/lib/mywebct_manager_util.ph
```

の中に定義部分を見つけた。後は不要なテキスト入力フィールドを生成している部分を削除するだけである。この場合は 1004 行目から 1050 行目までをコメントアウトした。これで図 3 右のような表示となる。

3.3 ファイルアップロード時の文字コード自動変換

CE3.8 から日本語版の開発も米国で行われるようになったらしく、そのためか MyFiles にファイルをアップロードする際の日本語文字コード自動変換機能がなくなってしまった。このため、デザイナーはあらかじめ HTML ファイルを utf-8 で作成しておくか、アップロード後に一つずつ変換しなくてはいけなくなった。これは非常に面倒な手順であり、筆者らも講習会などで教官に説明するたびに心苦しい思いをして来た。そこで、文字コード自動変換機能を独自に実装できないかということ考えた。

日本語文字コードの自動変換は厳密にはできないが、出現するコードの頻度を統計的に判断することで、ほとんど実用上問題ないレベルで自動変換を実現しているツールが複数存在する。一般には `nkf` がよく利用されているが、今回われわれは `imdkcv` [2] というフリーソフトウェアを使用した。

WebCT ではコンテンツそのもの以外にも、小テストの問題データベース、メール・掲示板の添付ファイル、提出課題、シラバス、などなど多くの場所でファイルのアップロードが行われる。これらの処理はすべて、

```
{webctdir}/webct/generic/lib/dbm_upload_util.ph
```

で定義される関数 `get_contents_and_filename()` に落ちて来るので、ここで自動変換の処理を行えばよい。具体的には処理手順の最後の部分 (286 行目) に

```
require "quickpatch/patchlib.ph";
pal_upload_single_file( $tempfile ) if ( $content_type =~ /^text\/\// );
```

の二行を加えた。`pal_upload_single_file()` は今回作成した関数で、引数として渡されたファイルを utf-8 に変換する。`$content_type` にはアップロードするファイルの mime-type が入っている。上のコードだと、`text/plain` とか `text/html` の場合に変換処理を行うことになる。

これだけでは、コンテンツを zip ファイルでアーカイブしてアップロードした場合に対応できない。zip ファイルでアップロードされるようなファイル数の多いコンテンツでこそ、自動変換機能が欲しいところだ。

ファイルマネージャで `unzip` 処理を行うと、

```
{webctdir}/webct/generic/lib/dbm_zip_util.ph
```

の `generic_unzip()` で実際の展開処理が行われる。一時的な作業ディレクトリに展開後、MyFiles に移動する処理をしている。ここでは、一時的な作業ディレクトリに展開した直後に、そのディレクトリに対する自動変換処理を付け加えた。`dbm_zip_util.ph` の 249 行目に

```
require "quickpatch/patchlib.ph";
pal_convert_directory ( $temp_dir );
```

の二行を加えた。`pal_convert_directory()` は、引数として渡されたディレクトリ内を再帰的に検索し、拡張子 `html`, `htm`, `txt` を持つものに対して utf-8 への変換処理を行うものである。

³require されているファイルは全部同一ディレクトリだったので、そのディレクトリで `grep` をかけた

patchlib.ph の内容を図 4 に示す。quickpatch は perl のインクルードパス (例えば `webctdir/per1/lib/perl5/5.6.1`) に置いた。pal_convert_single_file で meta タグの置換を行っているが、これは気分的なものである。WebCT では http のレスポンスヘッダで utf-8 が指定されるので、meta タグに charset の記述があつたとしても、まっとうな Web ブラウザでは無視されるはずである。しかし一致していないとなんとなく気分が悪いのでこうしている。

これまでのところ、この自動変換処理は大きな問題無く動作している。バックアップファイルからのリストア (内部的には zip ファイルの展開になる筈) で問題が生じないか気になったのだが、この処理は一般の unzip 処理とは別に行われているらしく、影響は無かった。

想定外だったのは、管理者が利用者登録をするためにアップロードするファイルも変換対象になってしまったことだ。われわれのサイトでは管理者のインターフェースを日本語 EUC にしていたため、問題が生じた。これは管理者のインターフェースを utf-8 にするか、もしくは管理者の作業だけは明示的に対象外にすることで回避できる。スクリプト内で環境変数 SCRIPT_NAME を参照すれば、どのスクリプトから呼び出されているかわかるので、管理者用スクリプトの場合は処理をしなければよいのである。

また、学生が課題としてテキストファイルを提出した場合にも変換されてしまうというのも微妙である。想定されている利用方法は、教官が課題をダウンロードして見ると言うもののようなので、utf-8 テキストファイルが手元 PC にそのまま来てしまったら処理に困る場合があるかもしれない。しかし一方提出書類がテキストの場合、WebCT 上で閲覧することもできるので、これが化けるのも困る。やはりダウンロードする際にも変換処理を入れないといけないうかもしれない。

4 おわりに

コンテンツの文字コードが自動的に統一され文字化けから解放されると言うのは、かなり気分がよい。是非 WebCT に加えて欲しい機能である。

本来なら元ファイルが utf-8 の時には変換処理をしないという判断が必要なのだろうが、imdkev では入力自動判別で入力が utf-8 で出力 utf-8 の処理を行っても特に問題が無かったためそのままにしている。nkf ではドイツ語を含む utf-8 文書をフィルタに通した際に文字落ちが生じた。これがあって、今回は imdkev を使用している。今後さらに他の言語を含むファイルでも検証を重ねないといけないだろう。

現在まだ実装をしていないが、ダウンロードの際にも変換処理を行うことを計画している。ダウンロードの場合にはどの形式のテキストにするかが問題になる。当初は MyFiles の直下にも設定ファイルを作って、そこに希望する文字コードを書いておくというように考えていたのだが、デザイナーの MyFiles は学生も使う。デザイナーと学生の計算機環境は同一とは限らない。ブラウザから得られる情報で判断するのが妥当かと考えている。

文献

- [1] 隅谷 孝洋, 稲垣 知宏, 長登 康, 中村 純 “広島大学における WebCT 運用” 第一回日本 WebCT ユーザカンファレンス, 2003
- [2] imdkev <http://www.d2.dion.ne.jp/~imady/freesoft/imdkev/>

```
sub pal_debug_str {
    my ($str) = @_;
    my ($datestr,@d);
    $datestr = `date "+%Y/%m/%d %H:%M:%S"`;
    chomp $datestr;
    open ( QUICKLOG, ">>/tmp/quickpatch.log" );
    print QUICKLOG ( "$datestr $str\n" );
    close ( QUICKLOG );
}

sub pal_convert_single_file {
    my ($path) = @_;
    my ($converter, $converter_argument);
    my (@content, $nometa, $line);
    my $cs = "utf-8";

    pal_debug_str ( "Converted: $path" );

    $converter = "/usr/local/bin/imdkev";
    $converter_argument = "-a -u8";

    (-x $converter) || return;
    (-w $path) || return;

    @content = ` $converter $converter_argument $path `;
    $content = join ( "", @content );
    $content =~
        s/(.*<meta[>]*charset=)(\w_-]+(?:[^\w-].*)/$1$cs$2/im;

    open ( PAL_OUT, "> $path" );
    print PAL_OUT $content;
    close ( PAL_OUT );
}

sub pal_convert_directory {
    my ($dir) = @_;
    my ($file,@files);
    (-d $dir) || return;

    @files = ` find $dir -name '*.htm' -or -name '*.html' -or -name '*.txt' `;
    chomp @files;
    foreach $file ( @files ) {
        pal_convert_single_file ( $file );
    }
}

# convert uploaded file to utf-8
sub pal_upload_single_file {
    my ($path)=@_;
    pal_convert_single_file ( $path );
}

1;
```

図 4: quickpatch/patchlib.ph の内容