Geodetic Data Modelling System: Honours Thesis

An online interface

Michael Still
964076

Geodetic Data Modelling System: Honours Thesis: An online interface

by Michael Still

Abstract

TODO

Table of Contents

Chapter 1. Introduction

Motivation

At the time that the **Geodetic Data Modeling System** (GDMS) implementation project was initiated, several of the potential users expressed a desire for the application to be available over the Internet. There are a variety of reasons that this type of functionality is enticing -- the main ones for GDMS are that it allows casual users to ability to analyse data, whilst not having to maintain their own copies of the datasets, and it allows users who would normally use the X windows interface to the application to access data whilst "in the field", or otherwise physically separated from their normal research location.

Initially, it was though that a Internet interface to GDMS was outside the achievable scope of the project for 2002. However, as the year progressed, it became clear to me that implementing a Internet interface was indeed achievable, and would add genuinely useful functionality to the application. Hence this honours extension was undertaken.

...

TODO

Chapter 2. Implementation rationale

Presentation of the user interface

In accordance with Internet user interface design best practise, the GDMS Internet interface presents a fully configurable user interface. This is implemented by providing extension tags to those which are available in the standard HTML specification (Raggett, Le Hors, & Jacobs 1999).

The GDMS Internet interface follows the recommendations of the most recent HTML specification at the time of development for the insertion of scripting tags into HTML source files (Raggett, Le Hors, & Jacobs 1999).

Chapter 3. Appendix One: GDMS Internet interface scripting elements

Introduction

This appendix documents the various tags which are implemented by the GDMS Internet interface in addition to the standard HTML tags normally available. All of these tags are within the scripting namespace provided by HTML 4.01 (Raggett, Le Hors, & Jacobs 1999).

commands

This tag will display a menu of the commands available in the current context -- for example, when the Internet interface first starts, it will list the open command. Commands which are not available, but are normally available will appear in a disabled style. This results in a list of commands which is consistent, and therefore reduces the potential for user confusion.

An example use is as follows:

```
&{commands};
```

The display of the command menu is altered by the following configuration file entries:

- *commandentry*: this HTML snippet is used for enabled commands
- *discommandentry*: this HTML snippet is used for disabled commands

For example, these configuration entries ship by default with the GDMS Internet interface:

```
# This line is simple HTML used for format the command entries,
# %s is the name of the command (including link HTML)
$commandentry = "<tr><td bgcolor=\"AAAAAA\">%s</td><tr>";

# This line is used for commands which aren't available...
$discommandentry = "<tr><td bgcolor=\"EEEEEE\">
  <font color=\"777777\">%s</font></td></tr>";
```

dataset

This tag will display the name of the current dataset, if one is open.

An example usage is:

```
&{dataset};
```

The dataset name is the portion of the filename before the extensions .dat1, .dat2, and .dat3 are applied. For example *mb_AUCK_GPS* is a dataset name.

datasets

This tag will display a list of the datasets available in the dataset directory.

An example usage is:

```
&{datasets};
```

The dataset directory is configured with the following configuration file entry:

```
# This line defines where the datsets are stored
$datasets = "/home/httpd/gdms-datasets/";
```

There are also several configuration options which alter the appearance of the list of datasets available. These are:

- *selectstart*: this is used for any output which should appear at the begining of the list. This could include HTML tags for the creation of the required list markup (for example tables).
- *selectentry*: this configuration item is used for each entry in the list. The special text **%s** is replaced by the HTML for the entry itself.
- *selectend*: this is used for any HTML required to finalize the list.

```
# This is used for selection lists (for instance datasets)
$selectstart = "List start";
$selectentry = "<LI>%s";
$selectend = "List end";
```

Chapter 4. Appendix Two: Source code

```
#!/usr/bin/perl

# GDMS web broker
# Copyright (C) Michael Still            2002
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

# This is the broker between the GDMS code and the CGI server. It’s role is to
# parse templates, execute requests, and cache results for speedy processing
# next time that request is seen...
use strict;
use CGI;

my($result, $command, $TEMP);

# Variables set by the config file
my($templates, $datasets, $commandentry, $discommandentry, $select-
start, $selectentry, $selectend, $plotcache, $tmpdir, $rooturl, $plo-
turl, $gdms, $temp);

# Setup the CGI module
$result = new CGI();
print $result->header;

# Read in the config file
eval ‘cat gdms.config‘ or die "GDMS web could not read it’s config file: $@";
print STDERR "Working directory is: ".‘pwd‘;
$temp = $gdms;
$temp =~ s/\/.*$//;
print STDERR "Changing to: $temp\n";

# Determine what page we are accessing
$command = $result->param(’command’);
if($command eq ""){
    $command = "main";
}

# Determine if the GDMS script for this page exists already (

print processTemplate("$templates/$command.html");
exit;



###############################################################################
# Find the template file, and then parse it
sub processTemplate(){
    my($file) = @_;
    my($pre, $post, $cmd, $output, $len, $line);

    # This local usage is used to make the TEMPLATE filehandle lo-
cal to this
    # subroutine...
    local *TEMPLATE;
    print STDERR "Processing template file: $file\n";

    $output = "";
```

```
   open TEMPLATE, "< $file" or
die "GMDS web could not open the template file $file";
   while(<TEMPLATE>){
# Repeatedly process a line until there are not more template commands
$line = $_;
$len = -1;

while($len != length($line)){
    $len = length($_);

    # todo: there must be a better way of doing this...
    $_ = $line;
    if(/(.*)&{([^%]*)};(.*)/){
 $pre = $1;
 $cmd = $2;
 $post = $3;

 if($cmd eq "commands"){
     # List the available commands
     $line = $pre.getCommands().$post;
 }
 elsif($cmd eq "dataset"){
     # The name of the current dataset
     $line = $pre.$result->param('dataset').$post;
 }
 elsif($cmd eq "datasets"){
     # List the datasets in the dataset directory
     my($temptotal, $tempfile);
     $temptotal = "";

     print STDERR "Getting datasets from $datasets\n";
     open TEMP, "find $datasets -type f -name \"*.dat1\" |";
     while(<TEMP>){
  my($linecount);
  $tempfile = $_;
  $tempfile =~ s/$datasets\/*//;
  $tempfile =~ s/.dat1\n$//;

  $linecount = `cat $datasets/$tempfile.dat1 | wc -l`;
  $temptotal = $temptotal.
      substHTML($selectentry,
         "<a href=\"$rooturl?command=main&dataset=$tempfile\">$tempfile<\/a> ".
         "($linecount lines)");
     }
     close TEMP;

     $line = $pre.$temptotal.$post;
 }
 elsif($cmd eq "motd"){
     # Output a message of the day
     $line = $pre.processTemplate("$templates/motd.html").$post;
 }
 elsif($cmd eq "xplot"){
     # A plot in the X direction
     print STDERR "Plotting in x direction\n";
     $line = $pre.generateAndLink("x").$post;
 }
 elsif($cmd eq "yplot"){
     # A plot in the Y direction
     $line = $pre.generateAndLink("y").$post;
 }
 elsif($cmd eq "zplot"){
     # A plot in the Z direction
     $line = $pre.generateAndLink("z").$post;
 }
    }
}

# And now we can print out the resultant line
$output = $output.$line;
   }
```

```
    return $output;
}

# Determine what commands should be available at this time
sub getCommands(){
    my($output, $temp);

    $output = "";
    if($result->param('dataset') eq ""){
 # Dataset open / close
 $output = $output.substHTML($commandentry, "<a href=\"$rooturl?command=open\">Open<\/
 $output = $output.substHTML($discommandentry, "Close");

 # Plotting
 $output = $output.substHTML($discommandentry, "Plot");
    }
    else{
 # Datset open / close
 $output = $output.substHTML($discommandentry, "Open");
 $output = $output.substHTML($commandentry, "<a href=\"$rooturl?command=main\">Close<\

 # Plotting
 if($result->param('command') eq "plot"){
     $output = $output.substHTML($discommandentry, "Plot");
 }
 else{
     $output = $output.substHTML($commandentry, "<a href=\"$rooturl?command=plot&datase
     $result->param('dataset')."\">Plot</a>");
 }
    }

    return $output;
}

# Substitute into the HTML stub from the config file
sub substHTML(){
    my($html, $insert) = @_;
    my($temp);

    $temp = $html;
    $temp =~ s/%s/$insert/;
    return $temp;
}

# This subroutine deals with generating plots as required and then outputs
# the HTML needed to link to that image
sub generateAndLink(){
    my($dir) = @_;
    my($file, $unique);
    local *COMMANDS;

    print STDERR "Started generateAndLink()\n";

    # Generate the filename
    $unique = "$$-".time()."-".rand();
    $file = "$plotcache/".$result->param('dataset')."-$dir.png";

    print STDERR "Filename is: $file\n";

    if(! -f $file){
 # We need to generate the image
 print STDERR "Plot cache miss for ".$result->param('dataset')." ($dir)\n";
 open COMMANDS, "> $tmpdir/gdms-$unique.cmd" or
     die "Could not open temporary file $tmpdir/gdms-$unique.cmd\n";
 print COMMANDS "open $datasets/".$result->param('dataset')."\n";
 print COMMANDS "plot $dir $file\n";
 close COMMANDS;

 # Execute the gdms main program with this command script
 `$gdms -b $tmpdir/gdms-$unique.cmd` or
```

```
            die "GDMS execution error for: $gdms -b $tmpdir/gdms-$unique.cmd";
 print STDERR "Return code as $?\n";
    }

    # Now link to that image
    return "<img src=\"$ploturl/".$result->param('dataset')."-$dir.png\">";
}
```

Code: eps

Chapter 5. Appendix Three: A sample configuration file

```
################################################################################
# Directories
################################################################################

# This line defines where the templates are stored
$templates = "/home/httpd/gdms-templates/";

# This line defines where the datsets are stored
$datasets = "/home/httpd/gdms-datasets/";

################################################################################
# HTML configuration
################################################################################

# This line is simple HTML used for format the command entries, %s is the name
# of the command (including link HTML)
$commandentry = "<tr><td bgcolor=\"AAAAAA\">%s</td><tr>";

# This line is used for commands which aren't available...
$discommandentry = "<tr><td bgcolor=\"EEEEEE\"><font color=\"777777\">%s</font></td>

# This is used for selection lists (for instance datasets)
$selectstart = "List start";
$selectentry = "<LI>%s";
$selectend = "List end";

################################################################################
# Caching configuration
################################################################################

# This is the location of the plot cache -- it can get quite big
$plotcache = "/home/httpd/html/gdms-plots/";
$ploturl = "/gdms-plots/";

# Location for temporary files
$tmpdir = "/home/httpd/gdms-temp/";

################################################################################
# Other
################################################################################

# This is the root URL
$rooturl = "/cgi-bin/gdms.pl";

# This is the full path to the GDMS application
$gdms = "/home/httpd/gdms";
```

Code: eps

Chapter 6. Appendix Four: References

References

Raggett, D., Le Hors, A., Jacobs, I., 24 December 1999 [last update], *HTML 4.01 Specification*, [Online] Available: http://www.w3.org/TR/html4/