# Linking GRASS with Chameleon

**Geo-analysis on the web**

*by Massimiliano Cannata*

## Introduction

Nowadays Web-GIS services are one of the most growing sector in the geographical information system science. It's popularity mainly reside in an easy to use interface for non-specialist to access geographical information in order to take decision. In most cases these web applications are focused on distributing geo-spatial information on the Internet in a "static" way: users can access and navigate the different maps, build combination of layer and print the results. More "dynamic" applications are now requested by different specialists: geologists ask for digitalisation tools, hydrologists ask for watershed analysis tools, and so on. Providing this kind of geo-analysis tools requires full access to typical GIS functionality like maps generation, map editing and map analysis. In this paper a new procedure for linking a package for deploying and managing Web mapping applications (Chameleon) and a package for geospatial data management and analysis (GRASS) is shown: such a procedure consist in the development of custom Chameleon's widgets accessing GRASS functionalities. Because of the FOSS environment of both this two softwares, this is a free, transparent and highly customised solution for developing "Web-GIS analysis tool".

## Linking GRASS with Chameleon

### Chameleon overview

Chameleon (DM Solution Group , 2006) is an highly customisable and adaptable environment for deploying and managing Web mapping applications. By using MapServer (University of Minnesota , 2006) as the back-end mapping engine that generates map images, manages mapped data and handles all of the geographic processing, Chameleon Web Mapping Components (CWC) provide an html-like tag system to incorporate in your current html pages the required mapping functionalities called widgets. In order to build a Web-GIS application with Camaeleon the developer needs to handle three files: the initialisation file (lunching the application), the mapserver mapfile (defining the layer's properties), and the Chameleon template (designing the look and the features of the application). This last file is the place where all the desired geographical tools (widgets) have to be inserted within the HTML, by using the defined tags and properties, in order to design the look of the Web interface. In the following lines a simple chameleon template file is shown as an example:

```
<html>
  <head>
    <title>CHAMELEON TEMPLATE</title>
  </head>
  <body onload="CWC2OnLoadFunction()">
    <form>
      <table>
        <tr>
          <td align="center">
            <CWC2 TYPE="MapDHTML"
                  VISIBLE="true"
                  WIDTH="400"
                  HEIGHT="300"
                  ALLOWRESIZE="true"
                  MARQUEECOLOR="#FF3333"
                  MARQUEEWIDTH="2"
                  MINSCALE="1"/>
           </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

Three elements can be noted : the FORM enabling the user-server interaction, the CWC2OnLoadFunction() enabling the processing of the argument passed by the FORM, and the CWC2 defining a Chameleon widget (in this case the map with the chosen options).

### GRASS call requirements

Three facts need to be considered in order to allow users to access GRASS functionality from the web in a safe way:

1. GRASS need a particular environment setting for running: this setting is based on particular environmental variables values and files and is generally set by the starting grass shell script.
2. GRASS allows just one user access for one MAPSET at a time. As GRASS has been designed for a desktop utilisation each MAPSET has his own "active" settings: multiple users could result in changing this values affecting all other running processes (i.e. two user using two different extension parameters for grass analysis).
3. Every web user needs to have his data protected (from other users) and available till the active PHP session is alive.

## Technical solution

To handle with these requirements a specific PHP class (grassOBJ) with the following methods has been developed:

- in the constructor most of the object variables are set;
- in the set_mapset_grassrc method a unique MAPSET and a unique .grassrc6 file (containing some GRASS variable) for the active PHP session are generated and set in the corresponding object variables;
- in the set_grass_env method all the system environment variable to be able to run GRASS are set;
- in the set_grass_region method the selected region is set as the active grass region.

A special Chameleon widget called SharedResource (that has no associated event but allows the setting of common variables) is used to specify the general GRASS variables values: an example of its structure is shown in the next lines.

```
1. <!-- Shared Resource for GRASS setting-->

2. <cwc2type="SharedResource" name="GRASSConfig">

3. <gisbase value="/usr/src/grass-6.0.0/bin"/>

4. <gisdbase value="/var/www"/>

5. <location value="NorthAmerica"/>

6. <projection value="epsg:2163"/>

7. <g_errors value="/var/www/msg/"/>

8. <g_messages value="/var/www/msg/"/>

9. </cwc2>
```

Where: line 2 open and define the sharedresource widget type and name; line 3 define the path to the binary grass commands; line 4 - define the DBASE path; line 5 define the LOCATION name; line 6 define the PROJ (epsg:XXXX) value for the grass LOCATION name (if not listed in the epsg file, just use the GRASS command ″g.proj -j″ and add the parameters in a new line with a new epsg number); line 7 define the path to the grass log error file; line 8 define the path to the grass log message file; line 9 close the sharedresource widget;
Taking advantage of the grassOBJ class new chameleon's widgets using GRASS functionalities (herein referred to as ″GRASS widget″) can be developed. Knowing that every widget object has the following base methods:

1. constructor - here the parent constructor is called and the attributes of the specific widget are defined;
2. InitDefaults, initialises the widget with the right values (the object variables are set);
3. ParseURL - reads the posted variables and executes the different functions (the core of the widget, where things happens);
4. DrawPublish - here the widget appearance (button, list, box etc..) is defined;

By adding the GRASSConfig shared resource as a parameter of a new GRASS widget all the needed setting will be available, therefore in the InitDefaults function a new grassOBJ object can be generated. Then in the ParseURL function the GRASS environment variable can be set by using the set_mapset_-grassrc and set_grass_env methods. In the same ParseURL function all the mapscript functionalities and the GRASS commands text variable needed can be generated and executed. The following lines show how a GRASS widget calling the r.stats command can be implemented.

```
class GrassStatistics extends CWCWidget {
 //generic variables
 var $moButton;
 var $moPopup;
 var $moLabel;
 var $layer;
 var $filename;

 //variables for Grass
 var $GrassConfigResource;
 var $oGRASS;
 var $config;
 ...
 // constructor
 function GrassStatistics() {
  // invoke constructor of parent
  parent::CWCWidget();
  ...
  $this->maAttributes['GRASSCONFIGRESOURCE'] =
  new StringAttribute('GRASSCONFIGRESOURCE',true);
  ...
 }
 function InitDefaults() {
  ...
  $this->GrassConfigResource =
  $this->maParams['GRASSCONFIGRESOURCE'];
  $config =
  &$this->maSharedResourceWidgets[ +
    $this->GrassConfigResource]->maszContents;
  $this->oGRASS =
  new grassOBJ($config);
  ...
 }
 function ParseURL() {
  ...
  // initialize env.variable for GRASS
  $this->oGRASS->set_mapset_grassrc();
  $this->oGRASS->set_grass_env();
  ...
  $cmd = "r.stats -acpln ".$this->layer;
  $cmd .= " output=".$this->filename;
  system($cmd)
  ...
 }
}
```

## Application example

In collaboration with Environmental Canada a WebGIS with watershed analysis tools has been developed in order to dynamically extract basins and theirs land use statistical property (HYDRO1k , 2006). The data used for this project are the "North American HYDRO1k data", in particular the DEM, the flow accumulation and the flow direction maps (HYDRO1k , 2006), 1 Km resolution; the "North America Land Cover Characteristics" from USGS (NALCC , 2006), 1 Km resolution; the Canada river measurement stations table information (converted in shapefile) and the Canada province boundary shapefile. For this application 4 new widgets using grass functionalities have been implemented (see figure 1), they allow the user to: extract a basin selecting an existing river station as outlet (GrassWatershedPoint); extract a basin selecting a generic point as outlet (GrassWatershedPixel); derive and show statistical information of the land use classes in the extracted basin with a plot of an histogram of density (GrassWatershedStatistics); and clear the extracted basin (GrassWatershedClear).

Figure 1: Chameleon widgets using GRASS functionalities (from left to right: GrassWatershedPoint, GrassWatershedPixel,GrassWatershedStatistics, GrassWatershedClear).

Used in conjunction with the ROI (Region Of Interest) widgets they provide a watershed analysis tool bar; standard widget are used in order to derive a navigation tools, a legend and some other functionality (print, help and error report). In figure 2 you can see a screenshot of the application where all these tools are clearly visible.

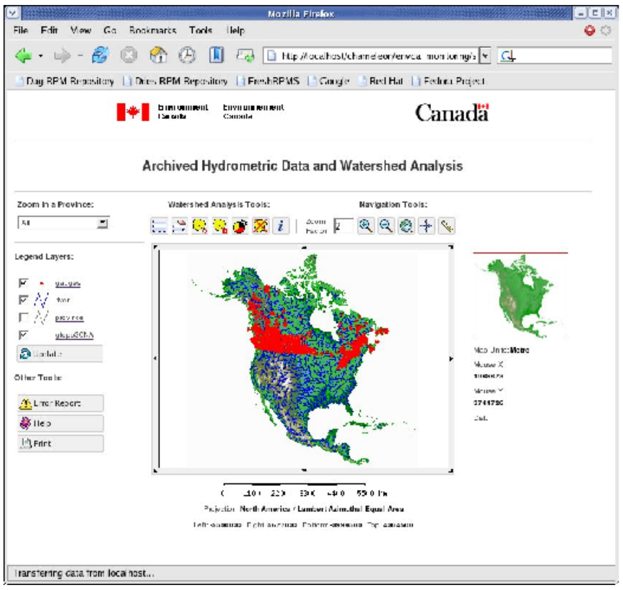

Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Canada
Archived Hydrometric Data and Watershed Analysis
Zoom in a Province:
Watershed Analysis Tools:
Navigation Tools:
Legend Layers:
Other Tools:
Error Report
Help
Print
Projection: North America Lambert Azimuthal Equal Area
Transferring data from localhost...

Figure 2: Screenshot of the example application.

As the rasters data to process is very large ( 73 million cells each), to prevent a too long response time of the server, as well as too big file generation, the GRASS principle of region has been used: a user in order to make analysis has to select a region with the ROI tools by drawing on the map a rectangle (see figure 3). *GrassWatershedPixel* or *GrassWatershedPoint* widgets will set this region (after reprojecting it in the current GRASS LOCATION projection) as the active analysis region in GRASS.

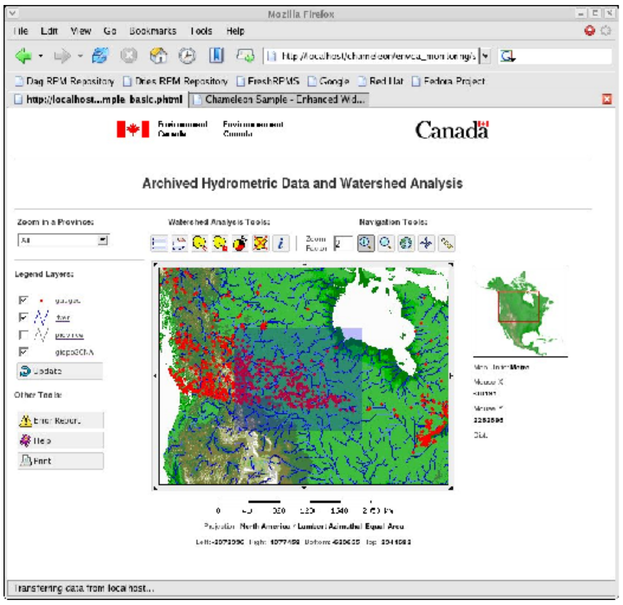

Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Chameleon Sample - Enhanced Wd...
Canada
Archived Hydrometric Data and Watershed Analysis
Zoom in a Province:
Watershed Analysis Tools:
Navigation Tools:
Legend Layers:
Other Tools:
Error Report
Help
Print
Projection: North America Lambert Azimuthal Equal Area
Transferring data from localhost...

Figure 3: Analysis area selection by using the ROI tools.

In case of use of the GrassWatershedPixel or GrassWatershedPoint widgets without a selected region an error dialog box is popped-up and the calcu-

lation is not performed.
Once a region is selected is it possible to calculate a basin by selecting a station or by selecting a pixel on the map, figures 4 and 5 show the basins calculated with both these methods.

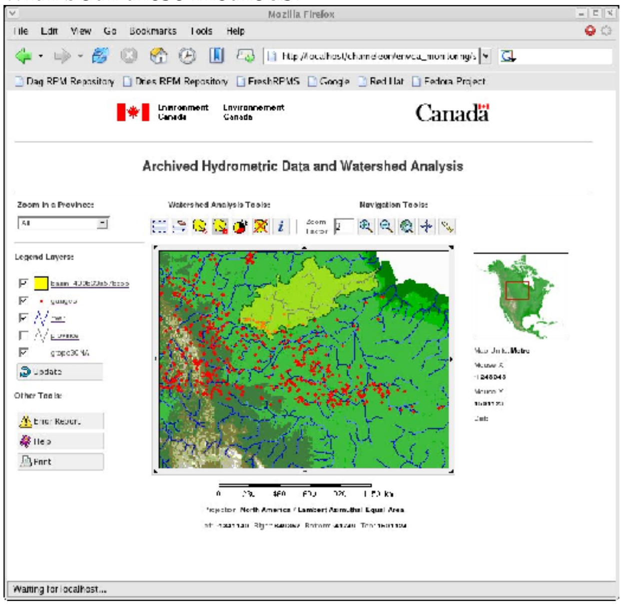

Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Environment Canada Environnement Canada
Canada
Archived Hydrometric Data and Watershed Analysis
Zoom in a Province:
Watershed Analysis Tools:
Navigation Tools:
Legend Layers:
Update
Other Tools:
Error Report
Help
Print
Waiting for localhost...

Figure 4: Basin calculated with GrassWatershed-Pixel.

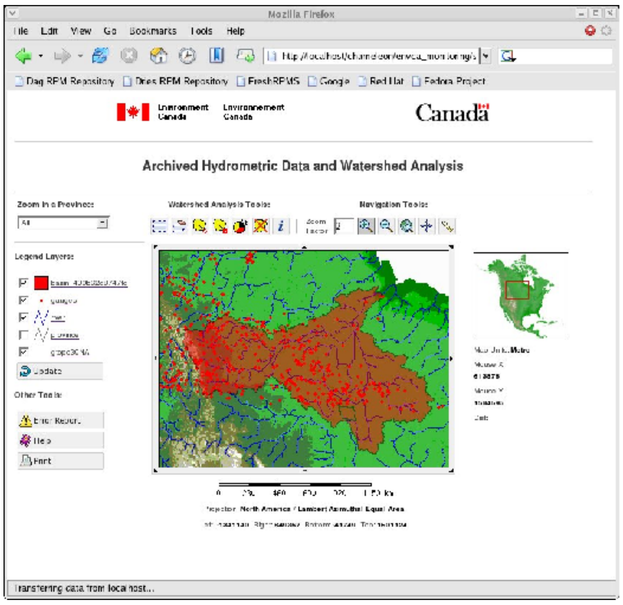
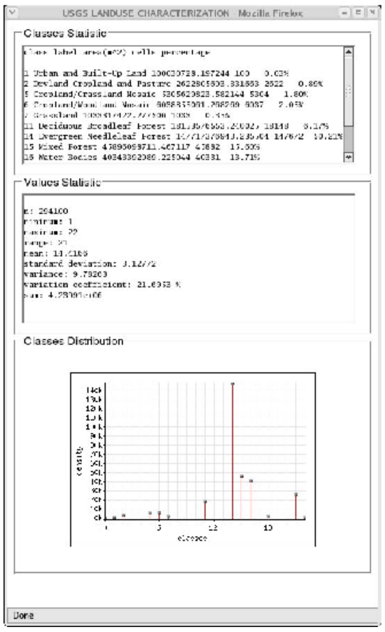

Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Environment Canada Environnement Canada
Canada
Archived Hydrometric Data and Watershed Analysis
Zoom in a Province:
Watershed Analysis Tools:
Navigation Tools:
Legend Layers:
Update
Other Tools:
Error Report
Help
Print
Transferring data from localhost...

Figure 5: Basin calculated with GrassWatershed-Point.

In order to avoid too small basin calculation a new GRASS command has been developed (r.nearest.coord). Given a coordinate pairs, a map, a threshold and a search radius it return the coordinate of the nearest cell in the raster within the specified radius having a value bigger or equal than the threshold. Because of a river network can be derived from a flow accumulation map by selecting the cells with a value higher than a threshold, if we use r.nearest.coord with such a map by using the the appropriate threshold value it will give us the coordinate of the nearest cell being on the river. This command, like the one for basin extraction (r.water.outlet), could result time consuming because of the high number of cells to handle: a key point in reducing calculation time is the use of an appropriate search radius parameter.
Once extracted a basin it is possible to calculate its statistical information based on another raster map by using the developed GrassWatershedStatistics widget: in the application a land use map is used. Once the widget is called a pop-up with three box is opened (see figure 6): the first box on the top shows the land use classes distribution (class number, class definition, area in square meters, number of cells and approximate percentage), the middle box shows some statistical parameters of the classes distribution (min, max, mean, standard deviation, etc.) and the bottom box shows a plot of the classes density (class vs. numerousity). All this values are related to the the current basin area calculated. Also in this case an error handling procedure is used and if this widget is clicked with no basin selected the boxes will show the warning message ”WATERSHED NOT SELECTED !”. The GrassWatershedClear widget simply allows a user to remove from the loaded map the basin calculated, as every new basin calculated has been load with a MapServer layer group value ”GRASS”: this command simply search and set the layer status property of these layers to MS_DELETE.



```
class label area(m^2) cells percentage

1 Urban and Built-Up Land 100030728.187244 100   0.03%
2 Dryland Cropland and Pasture 2622805503.331853 2622   0.89%
5 Cropland/Grassland Mosaic 5305620823.582144 5304   1.80%
6 Cropland/Woodland Mosaic 6088355081.208280 6087   2.07%
7 Grassland 1088817422.777500 1088   0.37%
11 Deciduous Broadleaf Forest 18135766552.240023 18148   6.17%
14 Evergreen Needleleaf Forest 147717378843.235304 147672   50.21%
15 Mixed Forest 43895098711.467117 43882   14.92%
16 Water Bodies 40343302089.225044 40331   13.71%
```

```
n: 294100
minimum: 1
maximum: 22
range: 21
mean: 14.4168
standard deviation: 3.12772
variance: 9.78283
variation coefficient: 21.6953 %
sum: 4.23991e+06
```

Figure 6: Watershed statistics calculated with the GrassWatershedStatistics widget.

## Conclusions

A link between Chameleon and GRASS has been generated, showing how a Web-GIS with analysis functionalities can be easily developed in a fully free and open source environment that guarantee an accessible system, both in terms of coding and costing. Analysis procedures require time to be executed: considering that these tools has been developed for specific users (and not for the mass) the resulting waiting time for the application is acceptable, being in the order of a couples of seconds (to give an idea GrassWatershedPixel processing time to analyse a region of 1492 cols * 1006 rows is 8 seconds and for a region of 604 cols * 652 rows it takes 3 seconds). Because this solution has not been thought to provide a full GIS server service, but just to add specific GIS capabilities to the Web application, in order not to overload the server itself, a restricted users accessing procedure should be considered, as well as other technicals solution like the selection of a limited region for raster analysis used in the application example.

## Acknowledgement

This work has been realized thanks to the funding of the ”DRIN - Dottorati di Ricerca, Industria e Nuova impresa” project by Fondazione Politecnico di Milano. Special thanks to Prof. Maria Antonia Brovelli (Politecnico di Milano), to the DM Solution Group staff (Ottawa, Canada), and to Environment Canada for their support.

## Bibliography

DM Solution Group (2006) *Chameleon* http://www.dmsolutions.ca/technology/chameleon.html

S. K. Jenson and J. O Domingue (1988) Extracting Topographic Structure from Digital Elevation Data for Geographic Information Systems Analysis *Photogrammetric Engineering and Remote Sensing* 54 (11): 1593-1600.

C. Ehlschlaeger (1989) Using the A\uT\d Search Algorithm to Develop Hydrologic Models from Digital Elevation Data *Proceedings of International Geographic Information Systems (IGIS) Symposium (Baltimore, MD, 18-19 March 1989)*: 275-281.

R. Blazek and L. Nardelli (2004) The GRASS Server *Proceedings of the FOSS/GRASS Users Conference 2004 - Bangkok, Thailand, 12-14 September 2004* http://gisws.media.osaka-cu.ac.jp/grass04/viewpaper.php?id=30

J. Y. Choi and B. A. Engel Real-Time Watershed Delineation System Using Web-GIS *Journal Computing in Civil Engineering* 17 (3):189-196

University of Minnesota (2006) *Mapserver* http://mapserver.gis.umn.edu

HYDRO1k Elevation Derivative Database (2006) *USGS* http://edcdaac.usgs.gov/gtopo30/hydro/namerica.asp

NALCC, North America Land Cover Characteristics (2006) *USGS* http://edcsns17.cr.usgs.gov/glcc/na_int.html

Hydro demo, watershed analysis demo site (2006) *IST* http://w3.ist.supsi.ch:8001/geomatica/

*Massimiliano Cannata*
*Institute of Earth Sciences (IST-SUPSI)*
*http://w3.ist.supsi.ch:8001/geomatica/*
massimiliano.cannata AT supsi DOT ch

# Simultaneous simulation of hydrological and carbon cycle processes in a GIS framework

**Integration of an existing distributed, process-oriented ecosystem model into GRASS GIS in combination with R**

*by Oliver Sonnentag*

## Introduction

The application of modeling techniques is a promising and widely used approach to many environmental problems and tasks in academia and industry, especially under circumstances in which direct measurements are not feasible and also for prediction purposes. Process-oriented models simulate physical processes based on fundamental principles, often with some degree of empirical generalization.
The simultaneous simulation of carbon cycle and controlling hydrological processes using a *distributed*, process-oriented ecosystem model such as the Boreal Ecosystem Productivity Simulator (BEPS) developed by Liu et al. (1997, 1999, 2002) is very data-intensive. A variety of software including GIS, remote sensing image processing, and statistical packages has to be employed to pre-process the required input data sets