Development of a \textit{p.mapper}-based webGIS

Luana Valentini
Politecnico di Milano – Polo Regionale di Como
http://geomatica.como.polimi.it
p.mapper (1)

- It is a framework based on MapServer and PHP/MapScript.
- It provides a good set of tools ready to use and it has a plugin API to add functionalities.
- Developed by Armin Burger, it is distributed under GNU GPL license:
  - it is released at no cost;
  - it is possible to modify the source code;
  - it is possible to use it for any purpose.
General view
p.mapper can be downloaded from:

http://www.pmapper.net/

both for Linux and Windows (MS4W package)

Some examples:

http://www.pmapper.net/gallery.shtml
Technology

Requirements:
- WebServer
- PHP5+ installed as CGI
- PHP/MapScript

\textit{p.mapper} makes use of:
- HTML
- CSS
- JavaScript
- XML
- PHP
Some functionalities

- Zoom/pan also via keyboard keys, mouse wheel, reference map, slider
- Easy configuration of layout and behaviour with XML config file
- Query functions (identify, select, search)
- Flexible layout of query results via JavaScript templates
- Query results display with database joins and hyperlinks
- Multilingual user interface pre-defined: EN, DE, IT, FR, NL, SK …
- HTML legends and various display styles of legend and layers/TOC
- Print functions: HTML and PDF
- Pop-up identify when moving with mouse over map
- Distance and area measurement function
- Adding points of interest with labels on map
- Plugin API to add custom functionality
- Various plugins: layer transparency, query result export, and more
Structure – MS4W (1)

**config**
The directory for the configuration files. Contains config_default.xml and subdirectory default, which is the default location for custom.js, js_config.php and the mapfile.

**images**
Contains all images used in the user interface. Subdirectory legend contains the legend icons. These are automatically created and updated.

**httpd_pmapper.conf**

Alias /pmapper "/ms4w/apps/pmapper/pmapper-4.0.0"

<Directory "/ms4w/apps/pmapper/pmapper-4.0.0">
AllowOverride None
Options Indexes FollowSymLinks Multiviews
Order allow,deny
Allow from all
</Directory>
javascript

Contain the JavaScript files compressed to a single file: 1 for standard p.mapper files `pm_cjs.js`, 1 for external libraries `xt_cjs.js` and 1 for all jQuery files `jquery_merged.js`. All uncompressed source files are under the subdirectory `/src/`. If you want to make modifications to an existing JS file it is recommended to do this in a separate JS file (like `custom.js`) under the `/config/.../` directory adding the modified function to this file. The utilities under `utils/compress_js` contain the script to compress all JS files.

JavaScript files:

* `pm.acommon.js`: commonly used functions, initialization of objects
* `pm.forms.js`: functions for forms and scale selection list
* `pm.geometry.js`: library for measurements/digitizing
* `pm.init.js`: init functions
* `pm.layout.js`: set layout parameters (width/height, left/top) of the GUI
* `pm.map.js`: main functions used for map creation, query, etc.
* `pm.pdraw.js`: map measuring/digitizing functions
* `pm.pmapper.js`: main p.mapper related functions, resize, GUI, dialogs, etc.
* `pm.query.js`: generic functions for queries
* `pm.search.js`: functions for attribute search
* ...
Structure – MS4W (3)

- **incphp**

  This directory contains the PHP files with the overall functions.

  * `common.php`: common functions and variables used in several php files
  * `custom.php`: customizing settings, at the moment mainly the definition of tools in the toolbar
  * `js_init.php`: initialize various JS variables from PHP
  * `/init/init.php & initmap.php`: initialization of the application
  * `initgroups.php`: initialization of GROUP and GLAYER objects
  * `/locale/language_*.php`: localization files for multilingual GUI
  * `map.php`: functions for rendering map
  * `/query/...`: functions for queries (identify, select, search)
  * `legend.php`: functions for creating legend
  * `group.php`: class functions for GROUP and GLAYER objects
  * `util.php`: auxiliary functions
  * `/print/print.php`: general print functions
  * `/print/pdfprint.php`: special print functions for PDF creation
  * `/print/tcpdf.php`: PDF creation functions
  * `/xajax/...`: PHP parts for XMLHttp/AJAX calls

- **plugins**

  Plugins for p.mapper are adding additional functionality and allow a more modulate design of the application.

  In addition, you can write your own plugin.

- **templates**

  Cascading Style sheets. Most formatting options in p.mapper are done via CSS
Same structure than in MS4W, in the directory
/var/www/
The main configuration for *p.mapper* is set under the subdirectory *config* in the file *config_default.xml*.

In this file you put the link at the *mapfile*.

Starting *p.mapper* can be done via the *map.phtml* file or a link from *index.html*. 
SHAPEPATH "../../data_como/shpf"
SYMBOLSET "../common/symbols/symbols-pmapper.sym"
FONTSET "../common/fonts/fontset.txt"

WEB
  TEMPLATE "map.phtml"
  IMAGEPATH "../../tmp/"
  IMAGEURL "/tmp/"
  METADATA
    "MAPFILE_ENCODING" "ISO-8859-1"
    "ows_onlineresource" "http://wms.yourserver.org?owskey=test&"
    "ows_srs" "EPSG:3003"
END
END
**Mapfile (2)**

- **p.mapper** supports **WMS/WFS OGC services**.
  We can simply add wms/wfs layers in the mapfile.

- **WMS example:**

```
LAYER
  NAME "DTM_20m"
  TYPE RASTER
  STATUS ON
  CONNECTION "http://www.cartografia.regione.lombardia.it/ArcGIS93/services/wms/dtm20_wms/MapServer/WMSServer"
  CONNECTIONTYPE WMS
  TEMPLATE "void"

  METADATA
    DESCRIPTION "DTM 20m"
    "wms_srs" "EPSG:3003"
    "wms_name" "0"
    "wms_server_version" "1.1.1"
    "wms_format" "image/png"

END
```

In this case we want the layer '0' in GaussBoaga coordinates (check in the GetCapabilities.xml file) and we will call this raster 'DTM 20m' in the webGIS.

[http://www.cartografia.regione.lombardia.it/ArcGIS93/services/wms/dtm20_wms/MapServer/WMSServer?REQUEST=GetCapabilities&Service=WMS](http://www.cartografia.regione.lombardia.it/ArcGIS93/services/wms/dtm20_wms/MapServer/WMSServer?REQUEST=GetCapabilities&Service=WMS)
Mapfile (3)

**WFS example:**

```plaintext
LAYER
  NAME "Railways"
  TYPE line
  STATUS off
  PROJECTION
    "init=epsg:32632"
END

CONNECTIONTYPE WFS
CONNECTION "http://wms.pcn.minambiente.it/cgi-bin/mapserv.exe?
  map=/ms_ogc/wfs/ferrovie_wfs_f32.map"
TEMPLATE "void"
METADATA
  "wfs_typename" "linee_ferroviarie_f32"
  "wfs_version" "1.0.0"
  "wfs_request_method" "GET"
  "wfs_connectiontimeout" "60"
  "wfs_latlongboundingbox" "486000 5055905 524999 5087904"
END

CLASS
  STYLE
    SYMBOL "rail"
    COLOR 51 51 51
    SIZE 2
END

http://wms.pcn.minambiente.it/cgi-bin/mapserv.exe?map=/ms_ogc/wfs/ferrovie_wfs_f32.map&service=wfs&request=getCapabilities
```

In this case we want the layer 'linee_ferroviarie_f32' in UTM/wgs84 coordinates (check in the GetCapabilities.xml file) and we use the 'rail' symbol to represent this layer.
In the LAYER object you have to add:

- **TEMPLATE "void"**
- **TOLERANCE 6**
- **METADATA**
  - **DESCRIPTION "Municipalities"**
  - **RESULT_FIELDS "NAME,POPULATION,SURFACE,DISTRICT"**
  - **RESULT_HEADERS "Name,Population [inhabitants],Surface [kmq],District"**
- **END**

We want to show in the query results only the fields 'NAME,POPULATION,SURFACE,DISTRICT', called in a different way ('Name,Population [inhabitants],Surface [kmq],District') - **optional**
If you insert an hyperlink in the attribute table and you want that this will be open when the user click on the query result table field, you have to add a line in the METADATA object:

```
METADATA
DESCRIPTION "Parchi Naturali"
RESULT_FIELDS "NOME_PAN,DATA_I,IDENT,LINK"
RESULT_HEADERS "Nome Parco,Data riconoscimento,Atto di riconoscimento,Link"
RESULT_HYPERLINK "LINK"
END
```

In this case when you click on the field 'LINK', that contains a link (e.g.: 'http://www.parcopineta.org'), pmapper open a new tab.

This is useful if you have completely different addresses that you want to link, otherwise you can use a JavaScript function:

```
METADATA
DESCRIPTION "Stations (GB - Roma40)"
RESULT_HYPERLINK "COMPANY"
RESULT_FIELDS "NAME,COMPANY,COMP_CODE,LINE"
RESULT_HEADERS "Name,Company,Company Code,Line"
END
```

When you click on company field a Wikipedia page about the train company is opened.
In the file `config/default/custom.js` we have to add:

```javascript
$.extend(PM.Custom,
{
    // Sample Hyperlink function for result window
    openHyperlink: function(layer, fldName, fldValue) {
        switch(layer) {
            case 'stations':
                if (fldName == 'COMPANY') {
                    window.open('http://en.wikipedia.org/wiki/' + fldValue);
                }
            break;
            default:
                alert('See function openHyperlink in custom.js: ' + layer + ' - ' + fldName + ' - ' + fldValue);
            }
        }
}
```

E.g.:

```
http://en.wikipedia.org/wiki/Ferrovie_dello_stato
```
<pmapper>
  <ini>
    <pmapper>
      <pmTitle>p.mapper - A MapServer PHP/MapScript Framework</pmTitle>
      <debugLevel>3</debugLevel>
      <plugins>export</plugins>
      ...list of the plugins that you want to insert
    </pmapper>
  </ini>
  <config>
    <pm_config_location>default</pm_config_location>
    <pm/javascript_location>javascript</pm/javascript_location>
    <pm_print_configfile>common/print.xml</pm_print_configfile>
    <pm_search_configfile>inline</pm_search_configfile>
  </config>
</pmapper>
...
...<map>
  <mapFile>como.map</mapFile>  map file that you want to use (in config/default folder)
  <tplMapFile>common/template.map</tplMapFile>  template map file
  <category name="cat_admin">
    <group>municipalities</group>
    <group>stations</group>
    ...
  </category>
  ...
  <allGroups>
    <group>Como_CTR</group>
    ...
  </allGroups>
  <defGroups>
    <group>lakes</group>
    ...
  </defGroups>
</map>
<layerAutoRefresh>1</layerAutoRefresh>
<imgFormat>png</imgFormat>
<altImgFormat>jpeg</altImgFormat>

<sliderMax>max</sliderMax>
<sliderMin>100</sliderMin>

</map>

<query>
<limitResult>300</limitResult>
<highlightColor>0 255 255</highlightColor>
<highlightSelected>1</highlightSelected>
<autoZoom>nquery</autoZoom>
...
</query>

<ui>
<tocStyle>tree</tocStyle>
<legendStyle>attached</legendStyle>
<useCategories>1</useCategories>
<catWithCheckbox>1</catWithCheckbox>
<scaleLayers>1</scaleLayers>
<icoW>18</icoW>
<icoH>14</icoH>
</ui>
config_default.xml: (4)

locale settings
(language, ...)

print formats

download dpi levels

php settings

export plugin formats

locale
  <defaultLanguage>en</defaultLanguage>
  <defaultCharset>UTF-8</defaultCharset>
  <map2unicode>1</map2unicode>
</locale>

print
  <printImgFormat>png</printImgFormat>
  <printAltImgFormat>jpeg</printAltImgFormat>
  <pdfres>2</pdfres>
</print>

download
  <dpiLevels>150</dpiLevels>
  <dpiLevels>200</dpiLevels>
  <dpiLevels>300</dpiLevels>
</download>

php
  <pearDbClass>MDB2</pearDbClass>
  <defaultTimeZone>Europe/Rome</defaultTimeZone>
</php>

pluginsConfig
  <export>
    <formats>XLS</formats>
    <formats>CSV</formats>
    <formats>PDF</formats>
  </export>
</pluginsConfig>
<searchlist version="1.0">
  <dataroot>$</dataroot>
  <searchitem name="municipalities" description="Municipality">
    <layer type="shape" name="municipalities">
      <field type="s" name="NAME" description="Municipality"  wildcard="0" />
    </layer>
  </searchitem>
</searchlist>
config_default.xml: (6) – search item

- Another example:

```xml
<searchitem name="population" description="Population &ge;">
  <layer type="shape" name="municipalities">
    <field type="n" name="POPULATION" description="Population &ge;" wildcard="0" compare="&ge;"/>
  </layer>
</searchitem>
</searchlist>
</ini>
</pmapper>
```
- PM.scaleSelectList = [10000, 25000, 50000, 75000, 100000, 125000, 150000, 200000]; → scale levels that you can visualize
- PM.queryResultLayout = 'table'; → layout of the query results
- PM.measureUnits = {distance:"[km]", area:"[km^2]", factor:1000}; → measure units
- PM.measureObjects = {line: {color:"#FF0000", width:2}};
config/default/js_config.php (2)

```php
PM.buttonsDefault = {
    toolbar: 'toolbar',
    options: {orientation: 'v',
              css: {height: '440px'},
              theme: 'default',
              imagetype: 'gif'},

    buttons: [
        {tool: 'space',
         dimension: 15},
        {tool: 'home',
         name: 'Zoom To Full Extent', run: 'PM.Map.zoomfullExt'},
        {tool: 'back',
         name: 'Back', run: 'PM.Map.goback'},
        {tool: 'fwd',
         name: 'Forward', run: 'PM.Map.gofwd'},
        {tool: 'zoomselected',
         name: 'Zoom To Selected', run: 'PM.Map.zoom2selected'},
        {tool: 'separator1',
         dimension: 1},
        {tool: 'zoomin',
         name: 'Zoom in'},
        {tool: 'zoomout',
         name: 'Zoom out'},
        {tool: 'pan',
         name: 'Pan'},
        {tool: 'separator2',
         dimension: 1},
        {tool: 'identify',
         name: 'Identify'},
        {tool: 'select',
         name: 'Select'},
        {tool: 'auto_identify',
         name: 'Auto Identify'},
        {tool: 'separator3',
         dimension: 1},
        {tool: 'measure',
         name: 'Measure'},
        {tool: 'pos',
         name: 'Add Point of Interest'},
        {tool: 'coordinates',
         name: 'Coordinates'},
        {tool: 'separator',
         dimension: 1},
        {tool: 'transparency',
        {tool: 'reload',
         name: 'Refresh Map', run: 'PM.Map.clearInfo'}
    ]
};
```

* Tool link elements

```javascript
PM.linksDefault = {
    containerid: 'toolLinkContainer',
    links: [
        {linkid: 'link',
         name: 'Link', run: 'PM.UI.showMapLink', imgsrc: 'link-w.png'},
        {linkid: 'print',
         name: 'Print', run: 'PM.Dlg.openPrint', imgsrc: 'print-w.png'},
        {linkid: 'download',
         name: 'Download', run: 'PM.Dlg.openDownload', imgsrc: 'download-w.png'},
        {linkid: 'help',
         name: 'Help', run: 'PM.Dlg.openHelp', imgsrc: 'help-w.png'}
//{linkid: 'layers',
    ]
};
```