
OSGeo Journal

The Journal of the Open Source Geospatial Foundation

Volume 1 / May 2007

In This Volume

Real World Implementations of Open Source software

Introducing Mapbender, deegree, openModeller ...

Understanding Spatial Relationships

Examining the Web Processing Server (WPS) Specification

Package Interaction - GRASS-GMT, Tikiwiki, PyWPS, GRASS-R ...

Software Updates

News, and more...



Integration Studies

Tikiwiki a GeoCMS

by Franck Martin

Overview

You may have heard of Content Management Systems (CMS), these are web based applications that allow you to share content over the web. The concept evolved by mixing wikis with forum based web sites. A wiki¹ web site allows fast creation of web pages using a simple formatting language. Links to new pages are first created, then the pages themselves. From this, forums, image galleries and blogs have been added. Tikiwiki² was created as an integrated CMS from the start as a more robust approach, allowing the interaction of all the features with each other. In Tikiwiki, you have a wiki, image galleries, file galleries, blogs, forums, articles, trackers and many more, all linked via a user and group management system to define permissions on objects. Tikiwiki is database independent, using *PHP ADODB*³ and it uses *Smarty*⁴, a template engine for PHP. It is then easy to personalise the look and feel of Tikiwiki so that your web site does not look like a default Tiki-

wiki site.

What is exactly Tikiwiki?

Tikiwiki is one of the first developed CMS, it is highly configurable but suffers from a large code base and a lack of modules. However, this disadvantage is also an advantage if you want to deploy all the functionalities in a CMS because they are tightly integrated rather than independent modules. Tikiwiki can also be extended via *mods*, a sort of plug-in system that allows you to extend functions. I chose Tikiwiki as a CMS because it was ranked as project of the month on Sourceforge and also because the developer community is very open. It was quite easy to add code to the code base. This is important when you start to contribute to a project.

Tikiwiki has an extensive security system and current development is focusing on implementing *AJAX* in the code to make it more efficient. The other strong point of Tikiwiki is internationalisation. There are several languages for the interface and it is easy to add your own language via the php equivalent of *gettext*. Also Tikiwiki supports the UTF8 character

¹Definition of wiki: <http://en.wikipedia.org/wiki/Wiki>

²Tikiwiki web site: <http://tikiwiki.org>

³PHP ADODB project: <http://adodb.sourceforge.net/>

⁴Smarty template engine: <http://smarty.php.net/>

set and right to left writing. You can mix several languages in the same page as well as provide integrated translation for wiki pages. Finally, Tikiwiki is highly customisable via its CSS and *Smarty* templates.

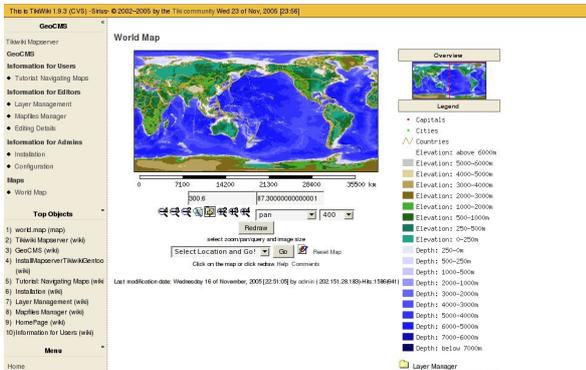


Figure 1: A standalone map in Tikiwiki.

A GeoCMS, What is it?

Since the success of Google Maps, maps are becoming an important feature of any web site. The success of Google Maps is due to the publication of its API, allowing any user to complement a Google map with point data, like the location of a user trip. Maps have been available for a long time on the web, like *Mapquest*, but were not available for customisation to the public. A *Geospatial Content Management System* (GeoCMS) is a CMS with geographical objects and map rendering capabilities. Despite having on your web site an online interactive web map, that you can zoom in, resize and query objects, you have also geographical objects that you can represent on the map. These maps can be standalone (see fig. 1) or included in a wiki page (see fig. 2). For instance, you can request your registered user to input their location and have them represented as a dot on the map. You can create an image gallery where each image has a location and have them represented as a dot on the map, to show your last trip, important monuments, geological features, and so on.

With any geographical layer there is the problem of knowing enough information about the layer (metadata): who created the layer, when, what is the quality of the data, in what projection system the objects are represented, etc.. It is important to have this information to build a level of trust in the data. If you have a population layer and a disaster strikes on a specific region, you must have relative trust in the

⁵MapServer web site: <http://mapserver.gis.umn.edu/>

⁶Tikiwiki installation documents: <http://doc.tikiwiki.org/>

data when preparing your response. Linking layers on the map with wiki pages is possible inside Tikiwiki. The metadata is therefore simple to enter and the specialist can then describe interesting features represented on the map.

Tikiwiki also provides an entire module to upload and process geographical data layers to put them together in maps via MapServer mapfiles. History is maintained for each mapfile, to keep track of changes and, optionally, alter you of such changes.

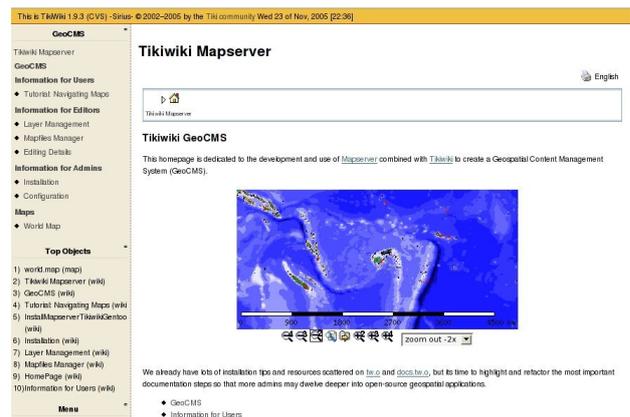


Figure 2: A map embedded inside a wiki page.

Installation and Setup

Tikiwiki maps feature has been made possible by the publication of the MapServer⁵ API for PHP. The MapServer engine is called within Tikiwiki to create images rendered inside the Tikiwiki interface.

I will not spend too much time on installation and setup, as the purpose of this article is to make you want to install it, so I will focus on describing its functionality, however I will quickly summarise.

Tikiwiki installation is quite straightforward and a complete installation documentation is available online⁶ for various systems including MS-Windows. There are some packages available for various Linux distributions as well. The Tikiwiki installation can be summarised to: untar and fill in a form that directs Tikiwiki to a blank database previously created. The installation of the *php-mapscript* extension from the MapServer code is somehow more complicated as you need to compile *MapServer* with all its dependencies and register the extension with PHP. You will also need to create a directory accessible by Apache and adequately protected to store mapfiles, geographical layers, and generated images. There

are also some packages available with some sample data, notably for Mandriva. For this specific installation the documentation can be found here:

<http://doc.tikiwiki.org/tiki-index.php?page=Maps+Install>

Creating maps

Tikiwiki uses the underlying MapServer engine, it means that it follows the mapfile format. This mapfile with the extension *.map* creates a web workspace to be used by the application. A text editor exists in Tikiwiki to edit the mapfile. It also uses quicktags to automatically fill some parts of the mapfile (for instance a template layer). There are several sections or objects in a mapfile. The overall object is the map object and starts with MAP and finishes with END. Inside this object you have the REFERENCE, LEGEND, WEB, SCALEBAR, and QUERYMAP objects. They define the overall representations of the map generated, the overview image (reference), the scalebar, legend and query highlights. Inside the MAP object you can insert LAYER objects. There are mainly two types of LAYERS: raster and vector. The use of the GDAL (pronounced goodle) library for raster and its included OGR (pronounced ogre) for vector allows read and write access to many Geographical Information Systems (GIS) file formats. The most commonly used formats are GeoTIFF for raster and Shapefiles or MapInfo TAB files for vector. Inside the LAYER object you define one or more CLASS objects to specify the color and the type of the feature drawn on the map. You can also add a LABEL object if you would like to label features. A typical mapfile can be found at <http://doc.tikiwiki.org/Maps+Mapfile+Tutorial>.

Each layer points to a GIS file which is also managed inside a layer management interface, allowing you to upload GIS files to the Tikiwiki site to be used inside mapfiles.

Tikiwiki takes advantage of comments and the METADATA object inside the mapfile. Tikiwiki places a header inside the mapfile, which allows it to keep track of revisions between each mapfile edition, a history is created as well.

⁷XAJAX library site: <http://www.xajax.org/>

⁸ka-Map site: <http://ka-map.maptools.org/>

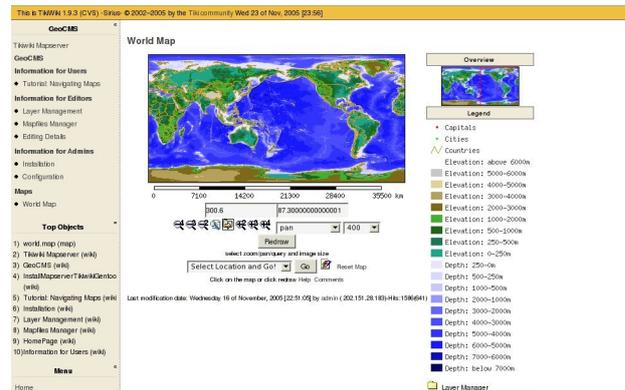


Figure 3: A table displays the information about the queried objects.

The METADATA keyword WIKI is used to link the layer to a wiki page inside wiki, allowing you to provide explanations on the specific layer. The link is created inside the layer manager on the map interface. The METADATA keyword DOWNLOAD is also used to allow the download of the GIS files composing the layer. This makes Tikiwiki a publication tool of Geographical data.

Finally the METADATA keyword VIEW is used to create pre-defined views of the map, so the user can zoom into the Pacific or Europe, or any other interesting place.

The system permits the inclusion of many layers, but Tikiwiki itself can create some specific layers. For instance, a layer containing the location of all the registered users which have specified a valid location can be created. In the same idea, an image gallery can be declared as geographic type and each image will receive a position. A layer out of all these images can then be created and added to any mapfile as points on the maps. When the user queries such a point, the table containing information about the image and the image itself is displayed just below the map (see fig. 3).

A wiki plugin has been developed which embeds any map inside a wiki page and then performs some basic function like zooming and panning. At its simplest level a generated image can be included inside any web page, including a wiki page, as all the parameters to generate a map image are included inside the URL. GET rather than POST is used to generate maps.

The new viewing interface

I have been working on is *AJAX*-based via the *xajax* library⁷ inside the standard viewing interface. But I am also proposing a new interface based on the ka-Map code⁸. I integrated this code inside Tikiwiki and it is provided as a *mods* (module) for Tikiwiki version 1.9.x and 1.10.x. As 1.10.x is still in development, it allows site manager to integrate the new interface immediately by downloading the *mod*.

The ka-Map code has been modified to make it portable, and a small library identifies ka-Map-ready mapfiles by looking for the keyword KAMAP inside the METADATA of the WEB Object. You just specify the name of the map and the various available scales, for instance:

```
KAMAP ?Pacific,1000000,500000,100000?
```

You can also use the standard ka-Map keywords as described in the ka-Map documentation. Finally the code identifies the LAYER METADATA keyword WIKI, and will display it next to the layer legend. The wiki page appears on top of the map with a semi transparent background. This new interface provides a Google Map look and feel for the navigation, while offering customisation of the data sets (see fig. 4).

The future is bright

Tikiwiki as a GeoCMS has been implemented in several countries⁹ and is becoming quite successful since MapServer PHP MapScript is quite stable with Apache. MapServer is an Open Geospatial Consortium (OGC) compliant Web Map Server (WMS), which means that many layers can be served over the internet to a wide range of map clients, providing total integration.

On the CMS side, there are several developments being considered. The first one is to include more javascript inside the map interface to show real-time map coordinates at the cursor position, to measure distances and to perform zoom functions based on rectangles drawn on the screen. In general, AJAX could be used to provide an even more user friendly interface.

Some additional functionality will be investigated and developed. For instance, the capability to locate geographical objects by their attributes, e.g.: Where is the town called Suva? For image layers and

other layers, the user should be able to hover the cursor over a point to display a box with all the attribute information at this location.

More objects in Tikiwiki will be made geographical, like file galleries, blogs, articles, etc. But the most interesting feature of Tikiwiki is to link trackers with maps. The tracker feature is, in fact, a system to create a form to enter data into a database. For the moment tracker data is only saved in the Tikiwiki database, but through the use of DSN the trackers could be saved in any database, including Postgresql. Postgresql through its PostGIS¹⁰ extension can store geographical objects and MapServer can read such Postgresql databases. Trackers would provide a way to enter any kind of geographical information. The extensions are then limitless including vehicle tracking, fish catch information, soil sampling systems, etc. The future is bright indeed and coders, documenters and users are always welcome.



Figure 4: Ka-Map Interface.

⁹South Pacific countries using GeoCMS: <http://www.sopac.org/maps>

¹⁰PostGIS site: <http://postgis.refractory.net/>



Figure 5: The Mapping Server system in Nauru.

Franck Martin

franck AT sopac.org

<http://www.peachymango.org/tiki-index.php?page=Franck+Martin>

Franck Martin is lead developer of the maps feature inside Tikiwiki. He works and lives in Fiji in the South Pacific and has been an earlier adopter of GNU/Linux. He is currently working on installing GeoCMS in 14 Pacific Island Countries (see fig. 5). This project is funded by the European Development Fund.

Editor in Chief:Tyler Mitchell - [tmitchell AT osgeo.org](mailto:tmitchell@osgeo.org)**Editor, News:**

Jason Fournier

Editor, Case Studies:

Micha Silver

Editor, Project Spotlights:

Martin Wegmann

Editor, Integration Studies:

Martin Wegmann

Editor, Programming Tutorials:

Landon Blake

Acknowledgements

Various reviewers & the GRASS News Project

The *OSGeo Journal* is a publication of the *OSGeo Foundation*. The base of this newsletter, the $\text{\LaTeX}2^{\text{e}}$ style source has been kindly provided by the GRASS and R News editorial board. All articles are copyrighted by the respective authors. Please use the OSGeo Journal url for submitting articles, more details concerning submission instructions can be found on the OSGeo homepage.

Newsletter online: <http://www.osgeo.org/journal>OSGeo Homepage: <http://www.osgeo.org>

Mail contact through OSGeo, PO Box 4844, Williams Lake, British Columbia, Canada, V2G 2V8

ISSN 1994-1897