The Open Source Geospatial Foundation

Open Source Tools for Geospatial Data Management
Tyler Mitchell

- Executive Director, Charter Member of OSGeo
- Author, *Web Mapping Illustrated* (O'Reilly, 2005)
- GIS Professional - Manager, User, R&D, Advocate
- Geographer
Open Source

“any program whose source code is made available for use or modification as users or other developers see fit...Open source software is usually developed as a public collaboration and made freely available.”
Open Source Examples

- Programming languages
- Linux operating system
- Apache web server
- Firefox web browser
- Thunderbird email client
- OpenOffice office suite
- Various databases
- GIS & mapping applications
- You name it...
Open Source Geospatial
Building on top of Open Source I.T.
Open Source I.T.

- Open Source as core I.T. infrastructure
- Focused on **communications**
- Trading **bits of text** & multimedia
- **Limited** ways to communicate
Open Source Geography

- Geospatial as interaction/planning tool
- Focused on measurable observations
- Trading geographic locations
- Unlimited interaction with geography
- *Where* matters to all - common foundation
Array of Geospatial Tools

• Typical needs and tools:
  data collection
  sharing data
  visualisation
  decision making
• Extending our senses
• Geospatial information management ties it all
Project Success & Quality

Quality depends on capability
  • Limited by your abilities & staff
  • Breadth of skills/experience required
Need agility and an eye to the future
  • standards
  • platforms
  • data issues
  • what's next?
  • beyond open source (OGC, etc.)
Software Stack
OSGeo Open Source Project Stack

- Desktop
  - GRASS GIS
  - gvSIG
  - OSSIM
  - Quantum GIS

- Metadata
  - GeoNetwork
  - Processing API
  - GEOS

- Web
  - Mapbender
  - MapBuilder
  - MapGuide
  - MapServer
  - OpenLayers

Data Access APIs
- FDO - GDAL/OGR - Geotools

Free Geodata
Project Stats

- From ohloh.net
- Missing gvSIG and deegree

* http://www.ohloh.net/stacks/151/report

Replacement Cost of your Stack

This calculator estimates how much it would cost to hire a team to write each project from scratch. The cost is the sum of each project cost. More »

<table>
<thead>
<tr>
<th>Include</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Codebase</td>
<td>5,450,649</td>
</tr>
<tr>
<td>Projects</td>
<td>14</td>
</tr>
<tr>
<td>Contributors</td>
<td>200</td>
</tr>
<tr>
<td>Effort (est.)</td>
<td>1474 Person Years</td>
</tr>
<tr>
<td>Avg. Salary</td>
<td>$55,000 year</td>
</tr>
</tbody>
</table>

$ 81,072,382
The OSGeo Software Stack

- **Data access libraries**
  
  GDAL/OGR - GeoTools – FDO - GEOS

- **Desktop GIS/mapping applications**
  
  GRASS - OSSIM - Quantum GIS - gvSIG

- **Web mapping clients & frameworks**
  
  Mapbender - MapBuilder - OpenLayers
  
  MapGuide Open Source - MapServer

- **Metadata catalog**
  
  GeoNetwork opensource

- **OGC Specs: WMS, WFS, WCS, GML, CSW...**
Data Access APIs

GDAL/OGR
Most widely used geospatial data access C++ library
50+ raster formats
20+ vector formats
Bindings in Python, C#, Java, Ruby, VB6, Perl

GeoTools
Java GIS toolkit
Serves data using various OGC specs.
Support for common geospatial data formats

Feature Data Objects (FDO)
GEOS
GDAL/OGR Libraries

- http://www.gdal.org
- GDAL – raster data access
- OGR – vector data access
- Available as programming libraries (C++, Python)
- Sample utilities are powerful
- Behind many products
GDAL Utilities

GDALINFO – get raster dataset info.

> gdalinfo globe.tif
Driver: GTiff/GeoTIFF
Size is 10800, 4800
Coordinate System is:
GEOGCS["unnamed",
   DATUM["unknown",
      SPHEROID["unnamed",6378137,298.2572235629972]],
   PRIMEM["Greenwich",0],
   UNIT[,0.0174532925199433]]
Origin = (-180.000000,90.000000)
Pixel Size = (0.00833333,-0.00833333)
Corner Coordinates:
Upper Left  (-180.000000,  90.0000000)
Lower Left  (-180.000000,  50.0000000)
Upper Right ( -90.0000000,  90.0000000)
Lower Right ( -90.0000000,  50.0000000)
Center      (-135.0000000,  70.0000000)
Band 1 Block=10800x1 Type=Byte, ColorInterp=Palette
   Color Table (RGB with 256 entries)
   0:  0,50,50,255
   1:  2,55,55,255
   2:  5,60,60,255
...
OGR Utilities

- **OGRINFO** – lists vector data info.

```shell
> ogrinfo testpoint.shp testpoint
INFO: Open of `testpoint.shp'
using driver `ESRI Shapefile' successful.

Layer name: testpoint
Geometry: Point
Feature Count: 3
Extent: (427446.218809, 6010049.544707) - (430765.476087, 6012780.281191)
Layer SRS WKT:
PROJCS["NAD_1983_UTM_Zone_10N",
    GEOGCS["GCS_North_American_1983",
        DATUM["North_American_Datum_1983",
            ellipsoid["North_American_1983",a=6371007.181,b=6377969.584,flat=1/298.266142,datum["North_American_Datum_1983",datum_era["1983"]],
            prime_meridian["Greenwich",datum_era["1983"]],
            unit[meter,foot]]],
    PROJECTION["Transverse_Mercator",
        false_easting=0.0,
        false_northing=0.0,
        central_meridian=-117.0,
        standard_parallel=39.0,
        unit[meter,foot]
    ]
]

Id: Integer (6.0)
pointname: String (10.0)
OGRFeature(testpoint):0
  Id (Integer) = 1
  pointname (String) = My Place
  POINT (427446.219 6012309.465)

>ogrinfo "PG:dbname=project1 host=localhost" mypointtable
```
GDAL Utilities

- GDAL_TRANSLATE – converts raster formats
  
  ```
  > gdal_translate -of "ECW" ortho.tif ortho_out.ecw
  Input file size is 800, 600
  0...10...20...30...40...50...60...70...80...90...100 – done.
  ```

- GDALWARP – re-projects, rectifies using GCPs, converts formats

- GDAL_MERGE.py – Python script for merging datasets together
OGR Utilities

• OGR2OGR – converts between OGR datasets
  > ogr2ogr output.shp input.dgn

• Defaults to ESRI shapefile format.
  > ogr2ogr -f "DGN" output.dgn testpoint.shp

• More complex
  > ogr2ogr -f "ESRI Shapefile"
    -where "type=3 or type=4" <-- use attributes in data
dgntoline.shp <-- output file
c:\temp\93j016.dgn <-- input file
-nlt LINESTRING <-- output data type
GIS & Image Processing

GRASS GIS

OSSIM Image Processing

gvSIG
Quantum GIS

(c) QGIS 2004
Land Cover

Land Cover Description
- Land Cover Type

Land Cover of Canada (image)
- Coniferous Forest: High Density (1)
- Coniferous Forest: Medium Density - Southern (2)
- Coniferous Forest: Medium Density - Northern (3)
- Coniferous Forest: Low Density - Southern (4)
- Coniferous Forest: Low Density - Northern (5)
- Broadleaf Forest (6)
- Mixed Coniferous Forest (7)
- Mixed Uniform Forest (8)
- Mixed Heterogenous Forest (9)
- Mixed Broadleaf Forest (10)
- Burns: Low Green Vegetation Cover (11)
- Burns: Green Vegetation Cover (12)
Community MapBuilder

OpenLayers

Mapbender
PostGIS

- Spatial objects for PostgreSQL enterprise database
- Open source
- GIS functions available
- Various binary packages available
- Accessible through many programs – SQL interface

http://postgis.refractions.net
Using PostGIS

• Various ways to access and load data
  - shp2gpsql command line tool
  - ogr2ogr

CREATE TABLE "mypointtable" (gid serial, "id" int4, "pointname" varchar);
SELECT AddGeometryColumn('', 'mypointtable', 'the_geom', '-1', 'POINT', 2);
INSERT INTO "mypointtable" (gid,"id","pointname",the_geom)
  VALUES ('0','1','My Place',GeometryFromText('POINT (427446.21 6012309.46)',-1) );
INSERT INTO "mypointtable" (gid,"id","pointname",the_geom)
  VALUES ('1','2','Your Place',GeometryFromText('POINT (430765.47 6012780.28)',-1) );
INSERT INTO "mypointtable" (gid,"id","pointname",the_geom)
  VALUES ('2','3','Work',GeometryFromText('POINT (430412.36 6010049.54)',-1) );
Select PostGIS data

• Using basic SQL

```
project1=# SELECT * FROM mypointtable;
gid | id | pointname | the_geo
0 | 1 | My Place | SRID=-1;POINT(427446.21 6012309.46)
1 | 2 | Your Place | SRID=-1;POINT(430765.47 6012780.28)
2 | 3 | Work | SRID=-1;POINT(430412.36 6010049.54)
```

• Analyse using PostGIS functions

SELECT buffer(the_geom,10) FROM mypointtable; → Returns a polygon
SELECT housenumber FROM houses, counties
  WHERE contains(counties.the_geom, house.the_geom); → houses in counties
SELECT transform(the_geom,4326)) from mypoints;
  → POINT(-124.113655717351 54.2535931617935)

• And much more…
MapServer Reads PostGIS

- Map file layer definitions can be for PostGIS data

```
LAYER
   NAME mypoints
   TYPE POINT
   STATUS ON
   CONNECTIONTYPE POSTGIS
   CONNECTION "host=localhost dbname=project1"
   DATA "the_geom from mypointtable"
CLASS
   NAME "Locations of interest"
   COLOR 0 0 0
   OUTLINECOLOR 255 255 255
   SYMBOL 'circle'
END
END
```
Application Support

- MapServer
- Quantum GIS (QGIS)
- ArcMap via PgArc extension
- uDig
- JUMP
Other Initiatives

• Project Incubator
• Educational Curriculum
• Public Geodata
• Promotion & Visibility
• Local Chapters - 20+
  Languages - Deutsch, Francophone, Español...
  Countries - Japan, India, China, Spain...
  Regions - Ottawa, New Mexico, South America...
How We Work

- Individual Volunteers
- Committees
- Project Steering Committees
- Board of Directors
- Funding
Invitation & Opportunity

• Practical ways to be involved
  - engage projects technically
• Strategic ways to show support
  - engage community
  - assist promotion
  - sponsorship
• Share your success stories
Contact

Tyler Mitchell, OSGeo
tmitchell@osgeo.org
http://osgeo.org
+1-250-303-1831
www.osgeo.org
wiki.osgeo.org

See you at FOSS4G 2008
Cape Town, end of September