

Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings

Volume 13 Nottingham, UK

Article 7

2013

OSM - GB: Using Open Source Geospatial Tools to Create OSM Web Services for Great Britain

Amir Pourabdollah

University of Nottingham, United Kingdom

Follow this and additional works at: <https://scholarworks.umass.edu/foss4g>



Part of the [Geography Commons](#)

Recommended Citation

Pourabdollah, Amir (2013) "OSM - GB: Using Open Source Geospatial Tools to Create OSM Web Services for Great Britain," *Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings*: Vol. 13 , Article 7.

DOI: <https://doi.org/10.7275/R5GX48RW>

Available at: <https://scholarworks.umass.edu/foss4g/vol13/iss1/7>

This Paper is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings by an authorized editor of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

OSM-GB

Using Open Source Geospatial Tools to Create OSM Web Services for Great Britain

by Amir Pourabdollah

University of Nottingham, United Kingdom.
amir.pourabdollah@nottingham.ac.uk

Abstract

A use case of integrating a variety of open-source geospatial tools is presented in this paper to process and openly redeliver open data in open standards. Through a software engineering approach, we have focused on the potential usability of OpenStreetMap in authoritative and professional contexts in Great Britain. Our system comprises open source components from OSGeo projects, the Open Street Map (OSM) community and proprietary components. We present how the open data flows among those components and is delivered to the Web with open standards. Apart from the cost issues, utilizing the open-source tools has offered some distinct advantages compared to the proprietary alternatives, if any was available. At the same time, some technical limitations of utilizing current open-source tools are described. Finally a case study is shown for the usability of the developed solution.

Keywords: OpenStreetMap, open source, open data, open standards.

Introduction

Since its inception in 2004, OpenStreetMap (OSM) has become the main free source of digital maps made by the crowd. Although OSM is rapidly growing in both contents and contributors, the belief that it is made by amateurs is perceived to limit trust in the value of this free data source within the traditional GIS community. The quality aspects of OSM have been investigated by different researchers and with different tools (Girres and Touya 2010; Haklay 2010; Zielstra and Zipf 2010; Mooney and Corcoran 2011). We contend that to encourage uptake of data, not only must the OSM community produce better maps but the authoritative and professional use cases of OSM shall be facilitated.

Because of the volunteer nature of the community, many open-source geospatial tools have been developed around OSM in the recent years, ranging

from data handling and data analysis to cartography and presentation. There are a number of core open-source tools that are used by the OSM developers, e.g. Mapnik (Pavlenko 2011) for rendering, while some other open-source tools have been developed for users and contributors e. g. JOSM (JOSM 2012) and the OSM plug-in for Quantum GIS (QuantumGIS n.d.).

Although those open-source tools generally fit the purposes of core OSM users and contributors, they may not necessarily fit for the purposes of professional map consumers, authoritative users and national agencies. If OSM is not effectively usable by this group, the gap between professional and amateur map producers/consumers may never be filled and a sense of trust in OSM may never happen among those users. On the other hand, if OSM can be used effectively by authorities, there will be a big chance that those users become active contributors, leading to even more usability and reliability.

Having observed this research gap, this paper presents a system with a strong reliance on open source geospatial components that can fit the British authoritative user's requirements. The solution is developed within the framework of a project called OSM-GB (OSMGB 2012).

In the rest of this paper, a software engineering approach is taken to analyze the specific requirements of authoritative users in Britain. Based on those requirements, the conceptual and detailed designs of our system are presented. The solution developed also benefits from open standards and open data initiatives as will be explained later.

Users' Requirements

In the Great Britain context, the following requirements were collected from authoritative users in the first stages of the project, particularly through steering group meetings and direct contacts. The requirements are a combination of general and GB-specific functionalities, as will be explained in this section.

It is however noticeable that gathering the users' requirements is done within the open-source/open-data contexts and implications, i. e. the users are generally willing to use the free software, data and standards as opposed to the licensed products (for many reasons including the known cost and updateness issues) but they are looking for the best possible solution that fits their specific purposes. Oth-

erwise, the designed system may not address the requirements from the users that wish to benefit from the contractual or the quality-guaranteed services offered by the available licensed products, e. g. the licensed map data from Ordnance Survey.

Coordinate Reference System

In the context of Great Britain, one of the main concerns in using a map in authorities is the availability of the map in the British National Grid (BNG or EPSG:27700) coordinate system. The OpenStreetMap website shows the map in Google Mercator (also called EPSG:900913), as is common in other public maps (e. g. Google and Bing maps). The OpenStreetMap tile service which can be used as the background in desktop GIS applications, is also only in Google Mercator. The export facility of OpenStreetMap website, as well as the OSM “planet files” and replications use OSM-XML format, in which the coordinates are encoded in Lat/Lon (EPSG:4326). There is other no free public service, at the time of this paper, which can deliver OSM data in the BNG coordinate system.

Multiplicity of Layers

Unlike public users, the authorities are specialized and need to have more flexibility in selecting the map features they require. They need the map to be thematically separated in layers, so the users do their job by superimposing the required layers. For vector maps, thematic filtering of the features is commonly feasible in GIS applications, but for raster maps such filtering is usually impossible. Although the features of OpenStreetMap are tagged and the tags can be used to separate them into multiple layers, the OSM website renders them all into a flat map. At the time of this paper, there is no free public service offering multi-layered raster map for OSM. It is however possible to use the filtering functionality of the OSM API, subject to knowing the required tag combination for the specified thematic map, which produces raw text output.

Raster and Vector Maps

Raster maps are necessary but not enough if detailed map analysis is required in the authoritative use cases. Querying and filtering by geometry or by attributes, applying routing algorithms, customized rendering and labeling are examples of the functionalities that can only be achieved in vector maps.

OSM-XML is not a known format in our target authorities. At the time of this paper, there is no other free public web service of vector OSM data for the GB area, particularly in BNG.

OGC Compliance

The standards established by Open Geospatial Consortium are the most widely accepted standards for map delivery via the Web. The availability of OGC compliant Web Services is a major advantage, if not a necessity, for a map service to be used by the authorities. The main OGC standards for map delivery are WMS (Web Map Service), WMTS (Web Map Tile Service), WMSC (Web Map Service, Cached) and WFS (Web Features Service). The first three standards are for raster map delivery and the last makes the vector text based output in a number of established OGC standards including GML (Geographical Markup Language) and KML (Keyhole Markup Language).

WMTS is currently run by the OpenStreetMap and some other mirrors in limited CRSs. Besides that, at the time of this paper, the only available world-wide WMS service is run by the University of Heidelberg called World OSM-WMS (GIScience 2010), which is single-layer and is updated weekly. No other WFS based on OpenStreetMap has been found for the GB area.

Consistent Access to National Open Maps

In Britain, Ordnance Survey provides a set of open data (OS 2013) in different formats and resolutions. The authorities may have full access to the Ordnance Survey’s open and non-open maps, however if they are willing to work with OSM they prefer to have an integrated access to those national maps in the same platform, particularly for superimposing the maps from different sources. The project’s focus has been on open data sets, thus integrating some parts of the OS Open Data with OSM and delivering them in a single platform can enhance the usability of the final product. Among the OS Open Data products, VectorMap District (VMD) is the most comparable to OSM and so was selected to be delivered via the OGC Web Services (WFS in this case) together with OSM. At the time of this paper, no free WFS access to the VMD is found (although the University of Edinburgh’s Digimap OpenStream service offers WMS access to OS Open Data, free for UK academic use).

Frequent Updates

The other main concern of the authoritative users is the frequency of updating the map information. A main advantage of crowd-sourced maps like OSM is the near-instant reflection of changes, in contrast to the long bureaucratic process of map updates in authority-sourced maps. After a user adds or changes data in OSM, the system is able to update its output in a very short time difference, perhaps in a few minutes. Any OSM redelivery service should therefore also update its content very frequently.

Quality Check and Fix

As a value-added feature, the OSM quality can also be checked in a defined geo-processing algorithm. Although the project has an extensive subsystem for data quality check and fix, this paper is not going to cover this subsystem in details. Particularly we do not cover the details of the user's requirements in this subsystem. It is however, an integrated part of the system design and the interactions of this subsystem with the rest of components are briefly mentioned.

Briefly, this functionality can benefit both the authoritative and public OSM contributors. The authoritative users can then have a gauge of OSM reliability, and the OSM contributors (including the authoritative users) can have guides to correct the map errors. The main requirement is to have a "rule-based" checking and fixing algorithm. It is not desired to have a closed hardcoded set of rules; instead, they shall be defined as openly and as flexibly as possible. The results of the quality checks and fixes are also desired to be delivered by some integrated and consistent Web Services.

It should also be noted that we are not claiming OSM to be a direct alternative to national mapping data such as that from the Ordnance Survey. The different sources have different characteristics: prime amongst these is the more precise (but closed) data capture & production specifications of the Ordnance Survey which guarantee certain levels of data quality in their data, particularly for example in completeness of coverage.

Summary

According to the items above, the current OSM website, its replications and redelivery mirrors are not likely to be effectively used by the authoritative map users in GB. A system is required that can deliver the

OSM data in BNG reference system, in vector and in thematic raster layers, by OGC-compliant means and with frequent updates. Finally it is desired to have consistent access to the OS Open Data and to have a quality checking subsystem.

The Conceptual Design

In order to fulfill the above requirements, a set of software components, including data sources, tools, databases and standards shall put together. Use of open-source tools has been a preference but not an absolute necessity in this project.

At the conceptual level of system design, Use Cases can show the interaction of the system with the external entities. In addition to the users who are the main actors, an automatic update scheduler can be considered as another actor. The OSM data source and its regular updates are the external data to be fetched regularly by the system. Our Use Case diagram is illustrated in Figure 1.

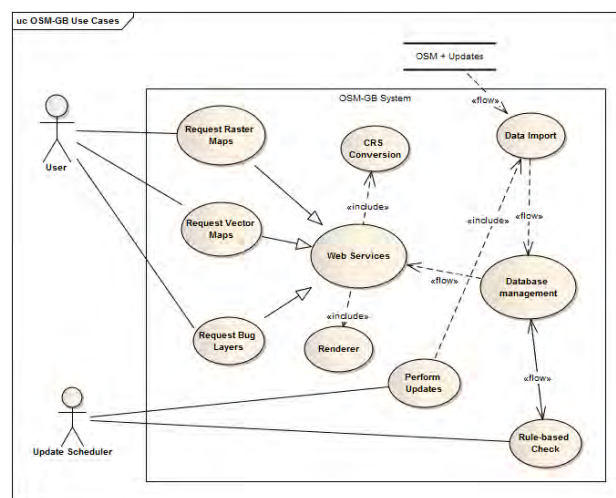


Figure 1: Use-case diagram for the conceptual level of design.

Users request the maps in raster or vector formats through the Web Services. In the case that they work with their preferred choice of GIS application, the application requests the information on their behalf. Moreover they may need to request for the detected "buggy" features which are yet another form of Web Service interaction. The Web Service provider takes its data from the central database management system, and also includes a CRS conversion activity. In addition, it needs to include a renderer component that can make cartographical outputs for the incom-

ing raster map requests.

The “Update Scheduler” actor frequently triggers an update signal. This signal primarily causes a new data import activity that updates the database with the new OSM replication; secondly it starts the checking algorithm that detects the “buggy” features. The rule-based check activity the necessary data is fetched from the database, checked against a set of quality checking rules; some fix on the data is applied and exported back to the database.

The Use Case diagram of Figure 1, helps to identify the necessary software components for each system’s feature. In summary, we need:

- A very flexible database management component that can effectively deal with spatial information
- Set of tools to efficiently import OSM and its updates into the database
- A powerful OGC compliant Web Service engine equipped with graphical renderer and the necessary CRS conversion means
- A rule-based quality check engine
- A workflow engine to orchestrate the frequent updating activities .

The Detailed Design

Data and Tools Diagram

Figure 2 shows the current configuration of the tools and the data flow inside the designed system.

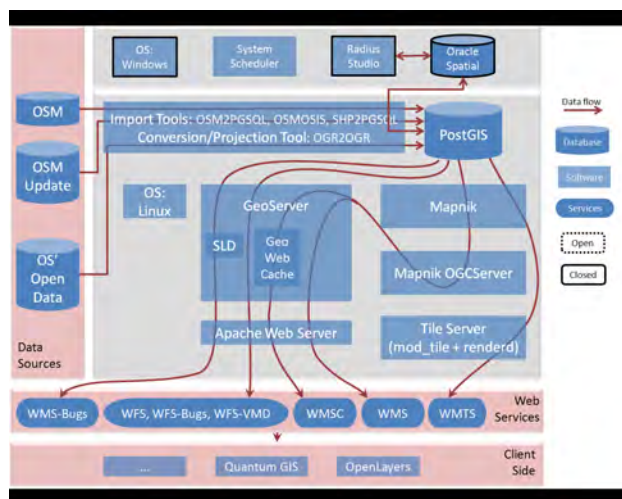


Figure 2: System Architecture; open tools, data and services.

As shown in figure 2, the current system mostly comprises open-source tools for its primary databasing and web services provision. It also shows that the data flow starts from the sources of open data and ends with the delivery in open standards and formats.

Open-source Tools and Databases

In the following subsections, the choices of open-source tools and databases are discussed. For each item, any available proprietary solution will be discussed.

PostGIS: The system databases are hosted in PostGIS 8.4.13 on Linux Ubuntu 11.0.4 platform. It has provided a very robust and stable functionality during the past year since the start of the project. The hosted databases include the whole GB processed data, daily differences, analysis data and the Ordnance Survey’s VectorMap District. The first database contains just about 3 million line features, 2 million polygons and 1 million Points of Interest (POIs). During the past years, although the daily workload of the database has been relatively heavy, there was not even one case of database crash.

The main alternative to PostGIS would be Oracle Spatial. We have had an instance of Oracle Spatial 11g available. However the limitations are firstly the efficiency of importing data from OSM-XML (using either open or proprietary tools), and secondly the efficiency of running spatial queries. In both cases, PostGIS has shown a very much better efficiency when tested. Moreover, the other limitation in adopting Oracle Spatial lies in the available OSM renderers.

OSM2PGSQL: The whole OpenStreetMap data file (also called the planet file) are available in OSM-XML format through a number of OSM mirror sites around the globe including Geofabrik (Geofabrik 2012). The data in this format includes all nodes, ways (a term used for lines and polygons in OSM) and relation features together with all of the associated attributes for each feature. On the other hand, the minutely, hourly and daily updates of the planet file are available in OSC (OpenStreetMap Changes) format via the OSM replication website (PlanetOSM n.d.). The OSC format is very similar to OSM-XML but excludes unchanged features, and the changed features are separated by the update type (create, change or delete). Each OSC file includes the change from its last replication time, whether a minute, an

hour or a day ago.

OSM2PGSQL (OSM-Wiki 2012) is an open-source tool that can import OSM-XML files into PostGIS database. It is also used to do the database update using OSC files. Its usability via the command line has made the simple documentation for the utility sufficient. In addition, the command-line usability has made this tool suitable to fit into workflows execution, compared with GUI-based alternatives.

There are some other tools that can import OSM-XML files into databases. OSMOSIS is an open-source tool that can do the import from OSM-XML to PostGIS but the database it produces is not directly usable by Mapnik.

Go Loader (Snowflake 2013) has been a non-open alternative. This tool has a recently added functionality that can import OSM-XML into Oracle Spatial database. However the process time is more than double of what is required by OSM2PGSQL. Moreover, it cannot work with OSC files so far. OSM2PGSQL has shown a very good performance compared with the alternative means studied for importing OSM-XML and OSC files.

Mapnik Library and OGCServer: Mapnik is the open-source tool for rendering the OSM data and is the default tool used by the OSM website. It uses the data stored in PostGIS and produces graphical output using a defined stylesheet. OGCServer is another open-source tool based on the core Mapnik library. It can make a server in order to wrap the graphics rendered by Mapnik in standard Web Services. We used the WMS output of this tool, which is by default able to make 90 separated raster layers out of source OSM data.

The main limitation we had in working with Mapnik was the documentation. For setting it up and for any customization, it has been difficult to find a complete documentation source. Individual developers' blogs, online forums and email lists have been mainly used, with the drawback of longer development and debugging time.

The alternatives have been very limited. OsmRender is another open-source tool but it is no longer maintained, is even less documented than Mapnik and also is less used by the developers because of its known projection bug (OSM-Wiki 2012). TileMill by MapBox (MapBox n.d.) is a more user friendly cartographic system in which users can define their own stylesheet in a graphical user interface. TileMill in fact uses Mapnik as its renderer core. We therefore decided to use Mapnik in this project to reduce any unnecessary wrapping or overhead around the core rendering engine, particularly because we

did not need to customize the rendering beyond the Mapnik defaults used on the OSM website.

Mod_Tile module and Renderd: These two open-source tools (OSM-Wiki 2012) are the front-end and back-end of the system's tile server (WMTS). They have nothing to do directly with the Mapnik OGCServer, since they serve map tiles directly through a different port.

Mod_tile is an Apache Web Server module that responds to tile requests and renderd is another component that handles the file system behind the tile cache. We used this combination to serve the tiles. Not only this is used in the "slippy map" front-end of the project website, but also this can be used for any client application that can use WMTS as the background map. Tirex (OSM-Wiki 2012) is another open-source alternative to renderd. Renderd is currently used by the OSM website which made it our first choice. The system will maintain the tile cache, allowing tiles to expire and be regenerated in response to data updates.

GeoServer and GeoWebCache: The Web Service functionalities are based on GeoServer. The embedded GeoWebCache component has also been used to manage pre-rendered, cached raster outputs.

GeoServer has the ability to wrap external WMSs which is particularly useful in our system. Although the Mapnik OGCServer makes a WMS server, wrapping it with GeoServer gave better service management. Particularly the WMS made by Mapnik OGCServer includes 91 layers (90 thematic layers plus a merged one) but by wrapping in GeoServer we could group layers and use custom projections. As a result, the 91 layers have been organized in 8 thematic groups: each has a number of sub-layers and are served in a customized list of projections. For example, "transportation" is a WMS layer group consisting of 40 sub-layers (different road types, bridges, tunnels, ferry routes, etc.). A user can then load any combination of the sub-layers or load the whole transportation layer in one.

Moreover, GeoServer can directly access the PostGIS database. As the result, a number of other Web Services have been created including raster and vector services for OSM, the OSM detected bugs and OS VMD. In total, the services produced by GeoServer include 2 WMSs, 1 WMSC, 1 WMTS and 3 WFSs.

We have started with GeoServer version 2.1.2 and later upgraded it to version 2.2.3. The issues we had in the older version was frequent crashes and memory leaks, limited projections and some bugs in GeoWebCache. Upgrading to 2.2.3 has shown a better stability, more integration and full support of the

BNG coordinate system in GeoWebCache. There are other open-source alternative for Web Services e. g. MapServer. GeoServer has been chosen for the ease of access and customization through its Web interface.

OSMOSIS: This open-source tool is a very general processing tool for OSM data (OSM-Wiki 2012). The pipe-line design of this command-line tool can do a variety of different jobs in importing, exporting, querying and filtering OSM data. In our system, this tool is used as a part of database updating process. OSMOSIS extracts and integrates the OSC (change files) from the OpenStreetMap replication website and makes it available to be used by OSM2PGSQL. This tool has also shown a stable performance during the project lifetime.

OGR2OGR and GDAL Library: This is an open-source command-line program that uses the GDAL library to import and export between different spatial file formats and coordinate systems (GDAL n.d.). We have used this to convert the coordinate reference systems when doing the import/export jobs for the PostGIS databases.

SHP2PGSQL: This is another open-source command-line program that comes with PostGIS installation. It converts ESRI's shape files into PostGIS database. This tool is used to import the Ordnance Survey's VMD into the database. VMD is openly published on CD-ROM as a series of shape files separated by the National Grids. SHP2PGSQL together with OGR2OGR is used to make a single database of the VMD in PostGIS that is used to serve the data through WFS.

Client Side Applications; OpenLayers and Quantum GIS: Although the standard outputs can be used by any OGC compliant client application, the focus of our system testing has been on two open-source tools: OpenLayers as the web interface tool and Quantum GIS as the desktop application. Both applications are able to work with WMS, WMTS, WMSC and WFS outputs. However, we mainly recommend OpenLayers for the light-weight services (WMTS and WMSC) and Quantum GIS for the heavier jobs (WMS and WFS).

We found a couple of issues in adopting these two applications with the other system's components: firstly in order to work with WMSC, OpenLayers uses the default value of 72 dpi for resolution (OpenLayers n.d.), while GeoServer uses OGC's 90dpi standard (GeoServer 2013). This shall be resolved by enforcing 90dpi in OpenLayers, otherwise the maps are not shown in the right position.

Secondly, requesting large vector maps can eas-

ily be very resource consuming. It is then useful to have an option of geometrically limiting the bounding box in a WFS request. In the WFS plug-in of Quantum GIS version 1.7 there is a checkbox for "Only request features overlapping the current view extend". However we found this option disappeared in the integrated WFS in version 1.8 which makes it impossible to work with large WFS datasets, unless the bounding box values are entered manually in the "WFS Filters" section.

Non-open-source Tools and Databases

As mentioned before, open-source tools have been a preferred but not a necessity when there is no better alternative. The sub-system of quality check and fix has a number of proprietary pieces of software. This has been because of the limitations in existing open-source software.

Radius Studio (1Spatial 2012) from 1Spatial is a powerful spatial information handling tool that has started its development since mid 90's. The established geo-processing power of this tool, especially on high computing power platforms, has been largely accepted within the community of professional spatial data users and it is used by a number of national mapping agencies, for example. Radius Studio internally uses a powerful object-oriented database that gives it the high performance in heavy geo-processing workloads. Moreover, all the system functionalities are web based and can be invoked by Web Services. In our view, there is a corresponding gap in the open source ecosystem, something that might be of interest to the open-source community.

Consequently, using a proprietary tool usually forces the utilization of some other proprietary tools. In this case, the Radius Studio's preferred OS is Windows and its preferred database is Oracle Spatial. As the result, we have been forced to have a separated server and database for the quality checking subsystem. Some parts of the two databases need to be synchronized and the proper conversion tools need to be used. The open-source OGR2OGR tool (mentioned earlier) can push the appropriate data between the two databases.

The Update Workflow

An interesting lesson learnt is in the central management of the two server sides (open-source and proprietary). We have Windows Server with proprietary tools and database on one side, and a set of open-source OS, tools and databases on the other side. On

which system should the system orchestrator be located?

This component shall be able to invoke different processes on the two sides. In our experience, external invocation of the open-source components is much easier than for the proprietary ones, at least for the tools we have used on the two sides. Thus our solution has been to locate the system scheduler on the Windows server. PostGIS, the importing and conversion tools installed on the Linux Server as well as Radius Studio sessions are all called from Windows Server. The System Scheduler is in fact a series of batch scripts that are controlled by Windows' Scheduled Task application. The details of the updating workflow are illustrated in figure 3, below.

As shown in figure 3, the data update workflow, which is currently run every day, invokes a series of database tasks on both servers. In each cycle, firstly the current status of database changes from the last run is backed up and then reverted back to what it was before the last round of quality fixes. Secondly the OSM changes (in OSC format) is downloaded from the OSM replication site and applied to the PostGIS database (using OSMOSIS and OSM2PGSQL tools).

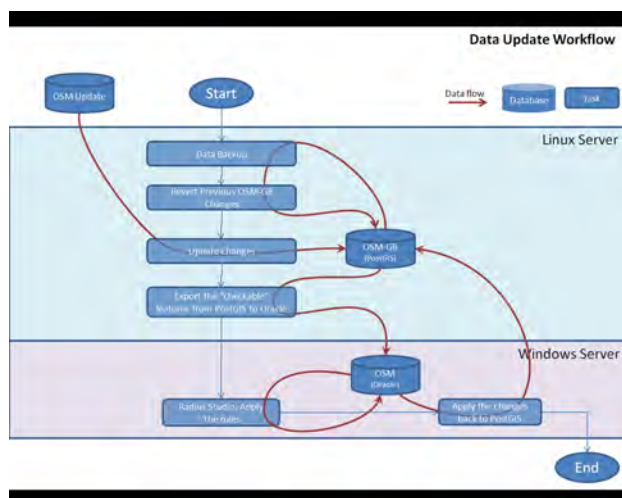


Figure 3: The data update workflow.

After the database is synchronized with the planet file, the database is ready to be processed according to the data quality rules. However, the database for GB is too big to be processed in less than a day and most of the data, which have not been updated, has already been processed in the previous run. Thus an algorithm (as PostGIS SQL commands) is run to select a subset of the OSM database which is needed to be processed by Radius Studio. This algo-

rithm has been optimized over the life of the OSM-GB project. Briefly, this subset includes all the added and changed features, all the features that have been "buggy" in the last run, and finally all of the features in a specified proximity of those two feature groups (currently a 20 meters buffer).

After the subset is selected, OGR2OGR is used to export the subset to the Oracle Spatial database. Radius Studio then runs a processing session and tags and/or fixes the buggy features. The results of the bug detection are later applied back to the PostGIS database, again using the OGR2OGR tool. The workflow timing depends on the hardware capacities, current size of OSM in the geographical extent, the amount of daily updates and the complexity of the applied rules and actions. Currently the workflow takes about 2 hours every day and there are about 100 thousands bugs reported and/or fixed in each cycle.

Open Data

The system is an example of both using and delivering data under open licenses. The data sources and the output data of the system are generally fit under the "open" term: the OpenStreetMap data and its regular updates are licensed under Open Data Commons Open Database License (ODbL), the Ordnance Survey's Open Data (OS 2013), of which VectorMap District is a part, is licensed under the UK's Open Government License for Public Sector Information. Finally the project shares its output under Creative Commons's (CC BY-SA 2.0) license.

Open Standards

The system is also a use-case of adopting the open standards from the Open Geospatial Consortium (OGC 2012). All the Web Services standards are provided in all three of the EPSG:4326, EPSG: 900913 and EPSG:27700 coordinate systems, in order to satisfy the user's requirements described in section 2. The full list of open standards used for the Web Services are:

WMS (Web Map Service): 90 transparent raster layers are categorized in 8 thematic groups and are served individually or in groups. The thematic groups are Land (5 sub-layers), Water (12 sub-layers), Buildings (7 sub-layers), Power (3 sub-layers), Boundaries (4 sub-layers), Transportation (38 sub-layers), Places (11 sub-layers) and Amenities (6 sub-layers). Besides the main WMS service, another WMS (called WMS-for-bugs) serves the detected er-

rors during the quality checking. The bugs are rendered and labeled according to the type of the detected bug. An issue in WMS was the compatibility between 1.1.1. vs 1.3.0 versions. The two versions specify the x and y coordinates in opposite order, so in our experience it is better the clients specify version 1.1.1 in the WMS request in order to get the right map positions.

WMSC (Web Map Service – Cached): This is a cached and single-layer version of the above WMS that optimizes the map delivery performance by keeping the pre-rendering tiles in a web cache.

WMTS (Web Map Tile Service): The single-layer 256 x 256 pixel tiles are served in OGC's WMTS standard request/response. The URL of getting a single tile contains z for zoom level and x, y for the tile's relative positions in the specified zoom level (e. g. http://www.osmgb.org.uk/osm_tiles/8/126/87.png).

WFS (Web Feature Service): This service makes the comprehensive vector information of OpenStreetMap in three layers (line, polygons, POIs). All the OSM key-value pairs for each feature are also included in the WFS response, which makes free filtering and querying possible. In addition to serving OSM, two other vector services exist: a WFS-for-Bugs which specifically serves the buggy features attributed with a description of each bugs, and a WFS-for-VMD which serves the imported OS VectorMap District data (for comparison purposes).

SLD (Styled Layer Descriptor): This OGC standard is used to compose the "Style" files in XML, used in rendering the buggy features in WMS-for-Bugs.

GML (Geography Markup Language) and KML (Keyhole Markup Language): These two OGC standards are among the variety of file formats used to encode the Web Services output, when applicable.

A Case Study

In this section a sample application of the usage of the developed service in an authoritative context is presented.

Surrey Heath Borough Council (<http://www.surreyheath.gov.uk>) uses a combination of Web mappings to manage its service delivery in the borough. The internal web mapping portal superimposes various map layers in British National Grid, including the maps sourced from Ordnance Survey. The WMS explained in this paper is readily used in this portal, particularly because it is available in BNG. Apart from being free, the benefit of using this layer is the "immediate" and "updateable" na-

tures of OpenStreetMap. When new constructions are developed in the area they can be immediately reflected in OSM and updated in less than a day. Moreover, the portal users can actively contribute to this update process, because they have access to other sources of information about the county's development projects.

Figure 4 shows overlaying the "waste collection points" on top of a national Ordnance Survey's MasterMap layer. As can be seen, many waste collection points are not associated with any house because the base map layer is not current with local house construction. Figure 5 on the other hand, shows the OSM-GB WMS layer used as the base map where the new houses are already mapped (centre and upper left). Thus the waste management team is using this map base to digitize new waste collection points.



Figure 4: Waste collection points (green and blue dots) on top of MasterMap.



Figure 5: Waste collection points (green and blue dots) on top of OpenStreetMap's WMS.

However there is an area in the OSM base map (lower right) that the houses are not mapped (though they are not new houses since they are already in MasterMap – Figure 4). In such a scenario, this is a very good motivation for the portal users to voluntarily contribute to OSM and map those houses.

Benefiting from the OSM short updating cycle is the core of what this case study can demonstrate. Particularly integrating the bug layers to the GIS application, can flag up the potential errors which can be fixed manually on the OSM if necessary, so the background maps can be easily updated. Moreover, it also demonstrates addressing the other requirements described in section 2: In terms of the data adaptation, OSM data is projected in the British National Grid so it can be easily integrated to the rest of the map sources in the system, either in raster or in vector. As far as OGC compliance concerns, the WMS/WFS services have made the OSM maps one of the many layers in the employed GIS application. Finally, the multiplicity of layers in WMS allows the user to focus on the required one (e. g. buildings or transportation here). More details of this case study can be found in (Rutter 2012).

Conclusion and the Future Works

In this paper we have presented an integrated open-source, open-data and open-standards solution for OpenStreetMap utilization in a professional context. A software engineering approach has been taken to gather the professionals' requirements from OpenStreetMap and to develop the solution having as much open-source components as possible. As evidenced by case studies, it has been shown that the current open-source tools that are already developed around OpenStreetMap can put together and configured to satisfy the professional's requirements. Apart from a few practical difficulties, the open-source tools has well been integrated, however there have been certain components like quality check and improvement that a non-open-source tool has been the only available choice.

The system has many rooms to expand as the future works. As with many other software life cycles, the requirements are dynamic and the system shall address them. Optimizing the performance, expanding the quality checking rules/actions and working closer with national map agencies are examples of those future works. Another future work is for the open-source geospatial community as suggested in the paper, to consider work on a rule-based geoprocessing tool for large-scale quality check/fix purposes. We also plan further work to promote the use of the bug reporting services in the OSM community for data quality checking.

Acknowledgement This research is based at the Nottingham Geospatial Institute at the University of Nottingham, and is funded and supported by 1Spatial and KnowWhere. We also wish to acknowledge the collaborations from Ordnance Survey GB, Snowflake Software and Pitney Bowes. The inputs and helps from James Rutter in Surrey Heath Borough Council are also highly appreciated.

References

- 1Spatial. (2012). 1Spatial Website. Retrieved December 2012, from <http://www.1spatial.com>.
- GDAL. (n.d.). ogr2ogr. Retrieved January 2013, from <http://www.gdal.org/ogr2ogr.html>.
- Geofabrik. (2012). Geofabrik. Retrieved January 2013, from <http://www.geofabrik.de/>.
- GeoServer. (2013). GeoServer 2.4.x User Manual - WMS Vendor Parameters. Retrieved January 2013, from <http://docs.geoserver.org/latest/en/user/services/wms/vendor.html>.
- Girres, J. F. and G. Touya (2010). Quality assessment of the French OpenStreetMap dataset. *Transactions in GIS* 14(4): 435-459.
- GIScience, U. o. H. (2010). WebMapService of World with OpenStreetMap-Data. Retrieved January 2013, from <http://www.osm-wms.de/>.
- Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and planning. B, Planning & design* 37(4): 682.
- JOSM. (2012). Java OSM editor Retrieved December 2012, from <http://josm.openstreetmap.de>
- MapBox. (n.d.). TileMill. Retrieved December 2012, from <http://mapbox.com/tilemill/>.
- Mooney, P. and P. Corcoran (2011). Integrating volunteered geographic information into pervasive health computing applications. *Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 2011 5th International Conference on, IEEE.
- OGC. (2012). Open Geospatial Consortium Web page Retrieved December 2012, from <http://www.opengeospatial.org/>
- OpenLayers. (n.d.). OpenLayers JavaScript Mapping Library Documentation. Retrieved January 2013, from <http://dev.openlayers.org/releases/OpenLayers-2.12/doc/devdocs/files/OpenLayers-js.html>.
- OS. (2013). Ordnance Survey OpenData. Retrieved January 2013, from <http://www.ordnancesurvey.co.uk/oswebsite/products/os-opendata.html>.
- OSM-Wiki. (2012). OpenStreetMap Project Wiki. Retrieved December 2012, from <http://wiki.openstreetmap.org/wiki/>
- OSMGB. (2012). OSM-GB Project Homepage - Measuring and Improving the Quality of OpenStreetMap for Great Britain. Retrieved December 2012, from <http://www.osmgb.org.uk>.
- Pavlenko, A. (2011). Mapnik. Retrieved December 2012, from <http://mapnik.org/>.
- PlanetOSM. (n.d.). Planet OSM. Retrieved January 2013, from <http://planet.osm.org>.

QuantumGIS. (n.d.). Quantum GIS Desktop Project. Retrieved December 2012, from <http://www.qgis.org/>.

Rutter, J. (2012). Is OpenStreetMapGB Useful? Damn Right it is! Retrieved January 2013, from <http://jamesrutter.posterous.com/is-openstreetmapgb-useful-damn-right-it-is>.

Snowflake. (2013). Go Loader. Retrieved January 2013, from <http://www.snowflakesoftware.com/products/>

[goloader/](#).

Zielstra, D. and A. Zipf (2010). A comparative study of proprietary geodata and volunteered geographic information for germany. 13th AGILE International Conference on Geographic Information Science.