

Testbed-12 WPS Conflation Service Profile Engineering Report

Table of Contents

1. Introduction	5
1.1. Scope	5
1.2. Document contributor contact points	5
1.3. Future Work	5
1.4. Foreword	5
2. References	6
3. Terms and definitions	7
3.1. Conflation	7
4. Conventions	8
4.1. Abbreviated terms	8
5. Overview	9
6. Background	10
6.1. Conflation	10
6.2. Hootenanny	10
6.3. WPS 2.0 Profiles	11
7. Using Hootenanny as Conflation backend in WPS	16
7.1. Concept	16
7.2. Implementation	18
7.3. Results	23
8. Conflation WPS Profiles	30
8.1. Conflation process concept	30
8.2. Generic Conflation Process Profile	30
8.3. Hootenanny Conflation Process Implementation Profile	31
9. Recommendations	36
Appendix A: Hootenanny Conflation Log	37
1.1. JSON Execute Request	37
1.2. Log	38
1.3. Summary	40
1.4. Statistical Aspects of the Input and Output Datasets	43
1.4.1. Summary of Basic Feature Elements	43
1.4.2. Summary of Input Features by Type	45
1.4.3. Summary of Output Features by Type	47
1.4.4. Summary of Area Features	50
1.4.5. Summary of Linear Features	52
1.4.6. Summary of POI Features	54
1.4.7. Summary of Unique Names	56
1.4.8. Summary of Tags	58
1.4.9. Summary of Translated Tags	60

1.5. Hootenanny Run-time Performance	63
Appendix B: Revision History	65
Appendix C: Bibliography	66

Publication Date: 2017-06-30

Approval Date: 2017-06-29

Posted Date: 2016-10-28

Reference number of this document: OGC 16-022

Reference URL for this document: <http://www.opengis.net/doc/PER/t12-A079>

Category: Public Engineering Report

Editor: Benjamin Pross

Title: Testbed-12 WPS Conflation Service Profile Engineering Report

Testbed-12 WPS Conflation Service Profile Engineering Report (16-022)

COPYRIGHT

Copyright © 2017 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is an OGC Public Engineering Report created as a deliverable of an initiative from the OGC Innovation Program (formerly OGC Interoperability Program). It is not an OGC standard and not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by

destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Abstract

One practical purpose of this ER will be to describe how a conflation tool such as the Hootenanny software can be used for conflation tasks using the Web Processing Service interface. The developed WPS REST (conflation) Service will be described in detail. Special focus will be laid on more complex conflation tasks that include user interaction. During earlier testbeds, we connected different conflation tools to the WPS and performed different conflation tasks (see [1] and [2]). The experiences gathered there together with the ones gathered in the Testbed 12 will be captured in the ER. As the WPS REST (Conflation) Service will be RESTful, this ER could be the basis for a REST binding extension for WPS 2.0. Service profiles are an important aspect of the WPS 2.0 standard. We will investigate how a WPS 2.0 Conflation Profile could look like in the hierarchical profiling approach of WPS 2.0.

Business Value

This ER will demonstrate how to use legacy software as a backend for WPS. The approach and any issues that might occur will help to better understand the requirements and methods that should be used to ease the use of legacy software with WPS. The second contribution of this ER should be the definition of a WPS 2.0 profile for conflation. This profile will be a validation of the approach described in the WPS 2.0 standard. Possible flaws will be detected and together with other profiles developed during Testbed-12 a best practice for developing WPS 2.0 profiles could be created.

Technology Value

The Conflation Profile that is described in this ER serves as proof of concept for the WPS 2.0 profiling approach. The implemented conflation process serves as use case for web-based processing.

Keywords

ogcdocs, testbed-12, conflation, WPS, profile

Proposed OGC Working Group for Review and Approval

WPS 2.0 SWG

Chapter 1. Introduction

1.1. Scope

This OGC® Public Engineering Report describes (1) a WPS process for conflating two datasets using the Hootenanny software and (2) WPS Conflation Profiles.

1.2. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Table 1. Contacts

Name	Organization
Benjamin Pross	52°North GmbH

1.3. Future Work

See section 9.

1.4. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

- OGC 14-065, OGC® WPS 2.0 Interface Standard

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9] shall apply. In addition, the following terms and definitions apply.

3.1. Conflation

Conflation is understood as the process of unifying two or more separate datasets, which share certain characteristics, into one integrated all-encompassing result.

Chapter 4. Conventions

4.1. Abbreviated terms

- API Application Program Interface
- OGC Open Geospatial Consortium
- OSM Open Street Map
- REST Representational State Transfer
- WFS Web Feature Service
- WPS Web Processing Service
- XML Extensible Markup Language

Chapter 5. Overview

This OGC® Public Engineering Report describes (1) a WPS process for conflating two datasets using the Hootenanny software and (2) WPS Conflation Profiles.

First, the concepts of the Web Processing Service and conflation will be introduced. Then the Hootenanny software will be introduced and the implementation will be described. Afterwards, the WPS profile for conflation will be described. Finally the findings of this ER will be summarized and recommendations will be given.

Chapter 6. Background

6.1. Conflation

Conflation, also known as map matching or merging is the process of combining two datasets based on a set of rules to produce a more complete dataset. One example is the combination of authoritative data, e.g. TIGER road data [1: <https://www.census.gov/geo/maps-data/data/tiger-line.html>] with crowd-sourced data, e.g. OpenStreetMap (OSM) [2: <http://www.openstreetmap.org/>]. The official data is collected with a high effort and follows strict requirements e.g. regarding data quality, which leads to a smaller frequency of updates. Crowd-sourced data on the other hand can be collected by virtually everyone e.g. by digitalizing streets from satellite imagery. The quality of the data is monitored by the community itself. Crowd-sourced data has therefore a high frequency of updates, with a trade off of having the danger of poor data quality for newly collected data. Several software systems exist to perform conflation tasks, e.g. the RoadMatcher software that was investigated in the OWS-9 testbed (see [1]). In Testbed-12 the Hootenanny conflation software was investigated regarding the feasibility for web-based conflation. The software is described in the following section.

6.2. Hootenanny

Hootenanny [3: <https://github.com/ngageoint/hootenanny>] is an open source conflation tool. It features the automated and semi-automated conflation of polylines, polygons and points. It internally uses the OpenStreetMap data structure. Hootenanny software offers a command line interface. Also, a webapp is provided including a Web interface to visualize the data. A WPS 1.0.0 interface is also provided to enable web-based conflation. The following data formats are supported by Hootenanny (source <https://github.com/ngageoint/hootenanny>):

Import: Hootenanny can ingest from:

- Shapefile (.shp)
- OpenStreetMap (.osm)
- ESRI File Geodatabase (.gdb)
- .zip files containing shapefiles and/or .gdb files
- geonames.org (.geonames)
- OSM API database sources (MapEdit, etc.; experimental feature; see documentation for workflow)

Export: Hootenanny can export to:

- Shapefile (.shp)
- OpenStreetMap (.osm)
- ESRI File Geodatabase (.gdb)
- Web Feature Service (WFS)
- OSM API database (MapEdit, etc.; experimental feature; see documentation for workflow)

6.3. WPS 2.0 Profiles

The WPS 2.0 Interface Standard [OGC 14-065] describes a hierarchical profiling approach for processes. The approach allows to harmonize process implementations to foster interoperability between WPS clients and servers from different vendors.

The highest level of process profiles are Process Concepts. They describe common principles among processes, e.g. buffering. Process concepts consist of a unique identifier and a descriptive document like a HTML page.

Generic Process Profiles are the next level of profiles and can be seen as abstract interface of processes. They consist of a detailed description of the process mechanics and define inputs and outputs, without the definition of concrete data exchange formats. Like the Process Concepts, Generic Process Profiles have a unique identifier and a descriptive document, e.g. describing the process in pseudo-code. Additionally, an XML document is needed following the GenericProcess schema [4: <http://schemas.opengis.net/wps/2.0/processProfile.xsd>]. The following listing shows an example generic process description:

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:GenericProcess
  xmlns:ows="http://www.opengis.net/ows/2.0"
  xmlns:wps="http://www.opengis.net/wps/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/2.0 ../wps.xsd">

  <ows:Title>Simple Features Buffer</ows:Title>
  <ows:Identifier>http://some.host/profileregistry/generic/SF-
Buffer</ows:Identifier>
  <ows:Metadata xlink:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/concept" xlink:href="http://some.host/profileregistry/concept/buffer"/>
  <ows:Metadata xlink:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/concept" xlink:href="http://some.host/profileregistry/concept/planarbuffer"/>

  <!-- Returns a geometry that represents all points whose distance from
        this Geometry is less than or equal to distance. Calculations are in the
Spatial Reference System of this Geometry. -->
  <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://some.host/profileregistry/generic/SF-Buffer.html"/>

  <wps:Input>
    <ows:Title>Input Geometry</ows:Title>
    <ows:Identifier>INPUT_GEOMETRY</ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://some.host/profileregistry/generic/SF-Buffer.html#input_geometry"/>
  </wps:Input>
  <wps:Input>
    <ows:Title>Distance</ows:Title>
    <ows:Identifier>DISTANCE</ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://some.host/profileregistry/generic/SF-Buffer.html#distance"/>
  </wps:Input>
  <wps:Output>
    <ows:Title>Buffered Geometry</ows:Title>
    <ows:Identifier>BUFFERED_GEOMETRY</ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://some.host/profileregistry/generic/SF-
Buffer.html#buffered_geometry"/>
  </wps:Output>

</wps:GenericProcess>

```

(source: [view-source:http://schemas.opengis.net/wps/2.0/xml-examples/profile-examples/SimpleFeaturesBuffer.xml](http://schemas.opengis.net/wps/2.0/xml-examples/profile-examples/SimpleFeaturesBuffer.xml))

The finest level of profiles are the Process implementation Profiles. They are process descriptions, defining also the data exchange formats. The following listing shows an example of a Process Implementation profile:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Process
  xmlns:wps="http://www.opengis.net/wps/2.0"
  xmlns:ows="http://www.opengis.net/ows/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/2.0 ../../wps.xsd">

  <ows:Title>Planar Buffer operation for GML features</ows:Title>
  <ows:Abstract>Create a buffer around a GML feature. Accepts any valid GML feature
and computes the joint buffer.</ows:Abstract>
  <ows:Identifier>http://some.host/profileregistry/implementation/Planar-GML-
Buffer</ows:Identifier>
  <ows:Metadata xlink:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/concept" xlink:href="http://some.host/profileregistry/concept/buffer"/>
  <ows:Metadata xlink:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/concept" xlink:href="http://some.host/profileregistry/concept/planarbuffer"/>
  <ows:Metadata xlink:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/generic" xlink:href="http://some.host/profileregistry/generic/SF-Buffer"/>
  <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://some.host/profileregistry/implementation/Planar-GML-Buffer.
html"/>
  <wps:Input>
    <ows:Title>Geometry to be buffered</ows:Title>
    <ows:Abstract>Geometry input in GML</ows:Abstract>
    <ows:Identifier>INPUT_GEOMETRY</ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://some.host/profileregistry/implementation/Planar-GML-
Buffer.html#input_geometry"/>
    <wps:ComplexData>
      <wps:Format mimeType="text/xml" encoding="UTF-8"
schema="http://schemas.opengis.net/gml/3.2.1/feature.xsd" default="true"/>
    </wps:ComplexData>
  </wps:Input>
  <wps:Input minOccurs="0">
    <ows:Title>Distance</ows:Title>
    <ows:Abstract>Distance to be used to calculate buffer.</ows:Abstract>
    <ows:Identifier>DISTANCE</ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://some.host/profileregistry/implementation/Planar-GML-
Buffer.html#distance"/>
    <wps:LiteralData>
      <wps:Format mimeType="text/plain" default="true"/>
      <wps:Format mimeType="text/xml"/>
    </wps:LiteralData>
  </wps:Input>
</wps:Process>
```



```

    <LiteralDataDomain default="true">
      <ows:AllowedValues>
        <ows:Range>
          <ows:MinimumValue>-INF</ows:MinimumValue>
          <ows:MaximumValue>INF</ows:MaximumValue>
        </ows:Range>
      </ows:AllowedValues>
      <ows:DataType
ows:reference="http://www.w3.org/2001/XMLSchema#double">Double</ows:DataType>
    </LiteralDataDomain>
  </wps:LiteralData>
</wps:Input>
<wps:Output>
  <ows:Title>Buffered Geometry</ows:Title>
  <ows:Abstract>GML stream describing the buffered Geometry.</ows:Abstract>
  <ows:Identifier>BUFFERED_GEOMETRY</ows:Identifier>
  <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://some.host/profileregistry/implementation/Planar-GML-
Buffer.html#buffered_geometry"/>
    <wps:ComplexData>
      <wps:Format mimeType="text/xml" encoding="UTF-8"
schema="http://schemas.opengis.net/gml/3.2.1/feature.xsd" default="true"/>
    </wps:ComplexData>
  </wps:Output>
</wps:Process>

```

(source: <http://schemas.opengis.net/wps/2.0/xml-examples/profile-examples/Planar-GML-Buffer.xml>)

Note the references to the Generic Process Profile in the metadata-element. Process profiles can be extended and extend other profiles. The following image shows the inheritance hierarchy for process profiles:

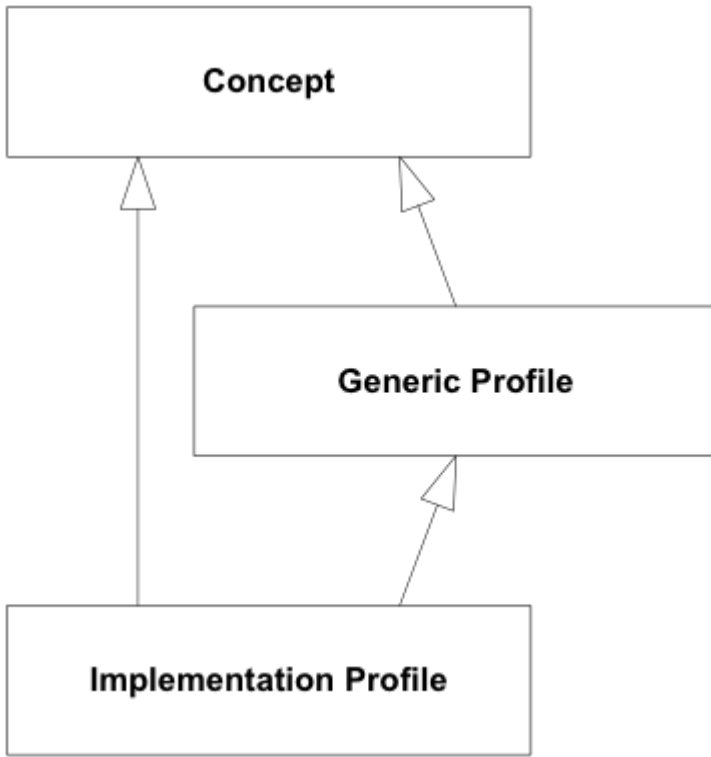


Figure 1. Inheritance hierarchy for process profiles (source [OGC 14-065])

The profiles for conflation are described in section 8.

Chapter 7. Using Hootenanny as Conflation backend in WPS

7.1. Concept

For high performance conflation tasks we will couple the Hootenanny software with WPS. Together with the conflation WPS service, a REST interface for WPS 2.0 was developed within Testbed-12. The following image shows the general architecture of the components.

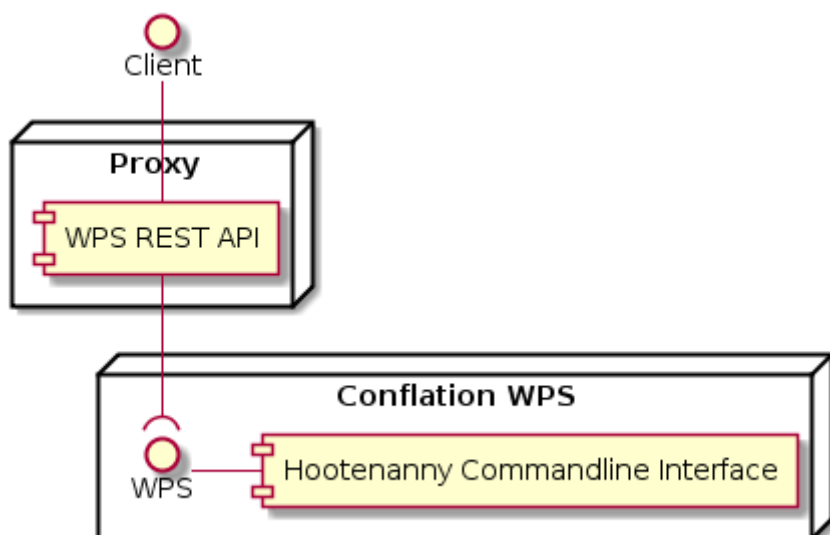


Figure 2. Conflation WPS Architecture

The REST interface is described in [OGC 16-035 - REST Architecture ER]. This section will focus on the WPS 2.0 process. To run the conflation with Hootenanny, we used the command line interface (see [4], section 14). In order to conflate two datasets, they need to be transformed to a modified OSM schema that is a superset of the original OSM schema [5: http://wiki.openstreetmap.org/wiki/Map_Features] (see also [4], section 2.1.2). For this transformation, so called translation files are needed that specify the mapping between attributes of the input files and attributes of the OSM schema (see [4], section 7). To explain the mapping, we use the example taken from the Hootenanny User Guide (see [4], section 14.29.2): The following table shows attributes for an example shapefile containing road data:

Table 2. Example road attributes

STNAME	STTYPE	FLOW
Foo St.	main	1
Bar Rd.	res	2

The columns are defined as follows:

- STNAME - The name of the street.
- STTYPE - The type of the street.
- FLOW - The flow of traffic, either 1 for one way traffic, or 2 for bidirectional traffic.

To map the attributes to the OSM schema used by Hootenanny, the following translation file could be used:

```
#!/bin/python

def translateAttributes(attrs, layerName):
    # Intialize our results object tags = {}

    # Is the STNAME attribute properly populated?
    if 'STNAME' in attrs and attrs['STNAME'] != '':
        tags['name'] = attrs['STNAME']
    # Is the STTYPE attribute properly populated?
    if 'STTYPE' in attrs and attrs['STTYPE'] != '':
        if attrs['STTYPE'] == 'main':
            tags['highway'] = 'primary'
        if attrs['STTYPE'] == 'res':
            tags['highway'] = 'residential'
    # Is the FLOW attribute properly populated?
    if 'FLOW' in attrs and attrs['FLOW'] != '':
        if attrs['FLOW'] == '1':
            tags['oneway'] = 'yes'
        if attrs['FLOW'] == '2':
            tags['oneway'] = 'no'

    # Return our translated tags
    return tags
```

Table 3. Road attributes matched to OSM schema

Original attributes	OSM attributes
{"STNAME":"Foo St.", "STTYPE":"main", "FLOW","1"}	{"name":"Foo St.", "highway":"primary", "oneway":"yes"}
{"STNAME":"Bar Rd.", "STTYPE":"res", "FLOW","2"}	{"name":"Bar Rd.", "highway":"residential","oneway":"no"}

Usage of the translation command:

```
ogr2osm [--limit n] (translation) (output.osm) (input1[;layer]) [input2[;layer]] ...
```

The OSM datasets can then be conflated using the conflate command.

Usage:

```
conflate (input1) [input2] (output) [--stats]
```

There are several configuration parameters available for the conflate command. We focused on the

following three parameters:

Table 4. Configuration parameters for the conflate command

Parameter name	Description	Value
conflate.stats.types	Type of conflation	String (allowed: reference, average)
highway.match.threshold	The threshold for calling a relationship a match.	Double (default: 0.161)
highway.miss.threshold	The threshold for calling a relationship a miss.	Double (default: 0.999)

By appending `--stats` to the conflate command, statistics about the conflation are saved for later review. The output of the conflation will be in the OSM format. This can directly be returned or transformed to a format like shapefile of geodatabase. This can be achieved using the `osm2ogr` command.

Usage:

```
osm2ogr (translation) (input) (output) [nodeCacheCapacity] [wayCacheCapacity]
[relationCacheCapacity]
```

Tho export just in the shapefile format and avoid the rather complex `osm2ogr` command, the command `osm2shp` can be used.

Usage:

```
osm2shp [columns] (input.osm) (output.shp)
```

7.2. Implementation

For the implementation a 52°North WPS was used, which is a geoprocessing-framework written in Java. The WPS interfaces version 1.0.0 and 2.0 are supported by the framework. The WPS with Hootenanny as backend was deployed on a development server running the CentOS 6.8 operating system.

The following diagram shows the sequence of the WPS conflation process in detail:

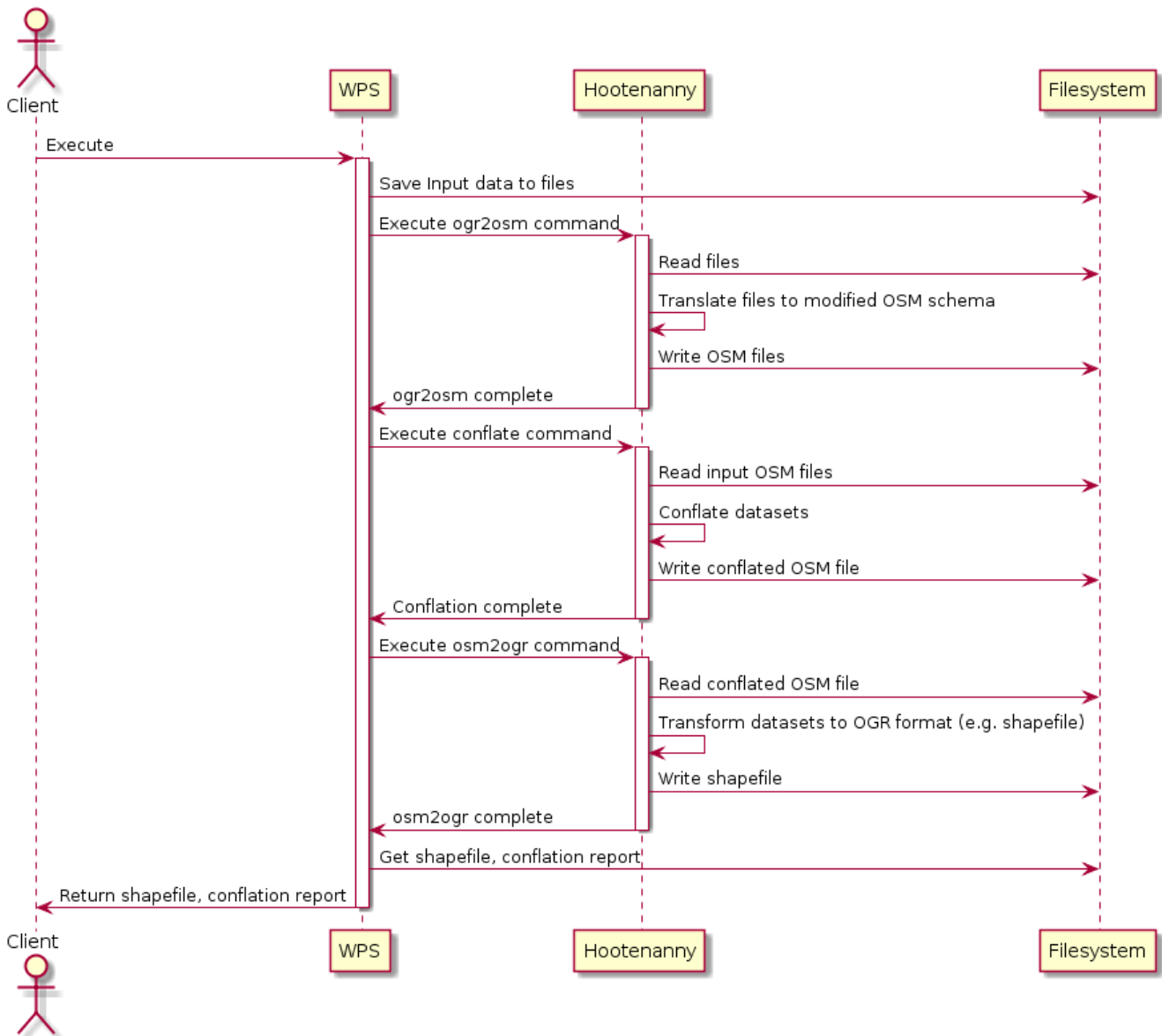


Figure 3. Conflation WPS Process Sequence Diagram

The input datasets that are sent to the WPS are stored as files in a temporary folder.

The WPS then makes calls to the Hootenanny commandline interface using the `java.lang.Runtime.exec(String command)` method. This method creates a Java process. The WPS waits until the process is finished and in case of success, it executes the next hootenanny command. In case of failure, the WPS process will stop and the WPS will return an exception report with details about the issue. The result of the conflation is transformed to a shapefile using the respective Hootenanny command. The shapefile is parsed by the WPS and converted to an internal data format. This additional step enables the WPS process to return the conflation result in various different formats like GML or GeoJSON.

The ProcessDescription of the process looks like the following:

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessOfferings xmlns:wps="http://www.opengis.net/wps/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/2.0
  
```

```

http://schemas.opengis.net/wps/2.0/wps.xsd">
  <wps:ProcessOffering processVersion="1.0.0" jobControlOptions="sync-execute async-
execute" outputTransmission="value reference">
    <wps:Process>
      <ows:Title>Hootenanny Conflation Process</ows:Title>
      <ows:Identifier>testbed12.lsa.HootenannyConflation</ows:Identifier>
      <ows:Metadata xlin:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/concept" xlin:href="http://52north.github.io/wps-
profileregistry/concept/conflation"/>
      <ows:Metadata xlin:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/generic" xlin:href="http://52north.github.io/wps-
profileregistry/generic/conflation"/>
      <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation"/>
      <wps:Input minOccurs="1" maxOccurs="1">
        <ows:Title>INPUT1</ows:Title>
        <ows:Abstract>Conflation input 1</ows:Abstract>
        <ows:Identifier>INPUT1</ows:Identifier>
        <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#input1"/>
          <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
            <ns:Format default="true" mimeType="application/x-zipped-shp"/>
            <ns:Format mimeType="application/x-openstreetmap+xml"/>
            <ns:Format mimeType="application/x-zipped-shp" encoding="base64"/>
            <ns:Format mimeType="application/x-zipped-gdb"/>
            <ns:Format mimeType="application/x-zipped-gdb" encoding="base64"/>
          </wps:ComplexData>
        </wps:Input>
        <wps:Input minOccurs="0" maxOccurs="1">
          <ows:Title>INPUT1_TRANSLATION</ows:Title>
          <ows:Abstract>Translation file for conflation input 1</ows:Abstract>
          <ows:Identifier>INPUT1_TRANSLATION</ows:Identifier>
          <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#input1_translation"/>
            <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
              <ns:Format default="true" mimeType="text/x-script.python"/>
              <ns:Format mimeType="text/plain"/>
            </wps:ComplexData>
          </wps:Input>
          <wps:Input minOccurs="1" maxOccurs="1">
            <ows:Title>INPUT2</ows:Title>
            <ows:Abstract>Conflation input 2</ows:Abstract>
            <ows:Identifier>INPUT2</ows:Identifier>
            <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"

```

```

xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#input2"/>
  <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
    <ns:Format default="true" mimeType="application/x-openstreetmap+xml"/>
    <ns:Format mimeType="text/xml"/>
    <ns:Format mimeType="application/x-zipped-shp"/>
    <ns:Format mimeType="application/x-zipped-shp" encoding="base64"/>
    <ns:Format mimeType="application/x-zipped-gdb"/>
    <ns:Format mimeType="application/x-zipped-gdb" encoding="base64"/>
  </wps:ComplexData>
</wps:Input>
<wps:Input minOccurs="0" maxOccurs="1">
  <ows:Title>CONFLATION_TYPE</ows:Title>
  <ows:Identifier>CONFLATION_TYPE</ows:Identifier>
  <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#conflation_type"/>
    <wps:LiteralData xmlns:ns="http://www.opengis.net/wps/2.0">
      <ns:Format default="true" mimeType="text/plain"/>
      <ns:Format mimeType="text/xml"/>
      <LiteralDataDomain>
        <ows:AllowedValues>
          <ows:Value>average</ows:Value>
          <ows:Value>reference</ows:Value>
        </ows:AllowedValues>
        <ows:DataType ows:reference="xs:string"/>
      </LiteralDataDomain>
    </wps:LiteralData>
  </wps:Input>
<wps:Input minOccurs="0" maxOccurs="1">
  <ows:Title>MATCH_THRESHOLD</ows:Title>
  <ows:Abstract>The threshold for calling a relationship a match. Defaults to
0.6. The higher the value the lower the TPR, but likely also the lower the
FPR.</ows:Abstract>
  <ows:Identifier>MATCH_THRESHOLD</ows:Identifier>
  <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#miss_threshold"/>
    <wps:LiteralData xmlns:ns="http://www.opengis.net/wps/2.0">
      <ns:Format default="true" mimeType="text/plain"/>
      <ns:Format mimeType="text/xml"/>
      <LiteralDataDomain>
        <ows:AnyValue/>
        <ows:DataType ows:reference="xs:double"/>
        <ows:DefaultValue>0.6</ows:DefaultValue>
      </LiteralDataDomain>
    </wps:LiteralData>
  </wps:Input>
<wps:Input minOccurs="0" maxOccurs="1">

```



```

<ows:Title>MISS_THRESHOLD</ows:Title>
  <ows:Abstract>The threshold for calling a relationship a miss. Defaults to
0.6. The higher the value the lower the TNR, but likely also the lower the
FNR.</ows:Abstract>
  <ows:Identifier>MISS_THRESHOLD</ows:Identifier>
  <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#match_threshold"/>
    <wps:LiteralData xmlns:ns="http://www.opengis.net/wps/2.0">
      <ns:Format default="true" mimeType="text/plain"/>
      <ns:Format mimeType="text/xml"/>
      <LiteralDataDomain>
        <ows:AnyValue/>
        <ows:DataType ows:reference="xs:double"/>
        <ows:DefaultValue>0.6</ows:DefaultValue>
      </LiteralDataDomain>
    </wps:LiteralData>
  </wps:Input>
  <wps:Input minOccurs="0" maxOccurs="1">
    <ows:Title>REFERENCE_LAYER</ows:Title>
    <ows:Abstract>The reference layer which will be dominant tags. Default is 1
and if 2 selected, layer 2 tags will be dominant with layer 1 as geometry snap
layer.</ows:Abstract>
    <ows:Identifier>REFERENCE_LAYER</ows:Identifier>
    <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#reference_layer"/>
      <wps:LiteralData xmlns:ns="http://www.opengis.net/wps/2.0">
        <ns:Format default="true" mimeType="text/plain"/>
        <ns:Format mimeType="text/xml"/>
        <LiteralDataDomain>
          <ows:AllowedValues>
            <ows:Value>1</ows:Value>
            <ows:Value>2</ows:Value>
          </ows:AllowedValues>
          <ows:DataType ows:reference="xs:integer"/>
          <ows:DefaultValue>1</ows:DefaultValue>
        </LiteralDataDomain>
      </wps:LiteralData>
    </wps:Input>
  <wps:Output>
    <ows:Title>CONFLATION_OUTPUT</ows:Title>
    <ows:Identifier>CONFLATION_OUTPUT</ows:Identifier>
    <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#conflation_output"/>
      <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
        <ns:Format default="true" mimeType="application/x-zipped-shp"/>

```

```

        <ns:Format mimeType="application/x-zipped-shp" encoding="base64"/>
        <ns:Format mimeType="application/x-zipped-gdb"/>
        <ns:Format mimeType="text/xml"
schema="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"/>
        <ns:Format mimeType="application/vnd.geo+json"/>
    </wps:ComplexData>
</wps:Output>
<wps:Output>
    <ows:Title>CONFLATION_REPORT</ows:Title>
    <ows:Identifier>CONFLATION_REPORT</ows:Identifier>
    <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#conflation_report"/>
        <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
            <ns:Format default="true" mimeType="text/plain"/>
        </wps:ComplexData>
    </wps:Output>
</wps:Process>
</wps:ProcessOffering>
</wps:ProcessOfferings>

```

The process implements the Hootenanny Conflation Process Implementation Profile described in section 8.3.

7.3. Results

To test the WPS conflation process, roads in the Area-Of-Interest (AOI) were conflated. The National Map (TNM) road segment datasets were used as reference layer. As geometry snap layer a recent snapshot of OSM data covering the AOI was used.

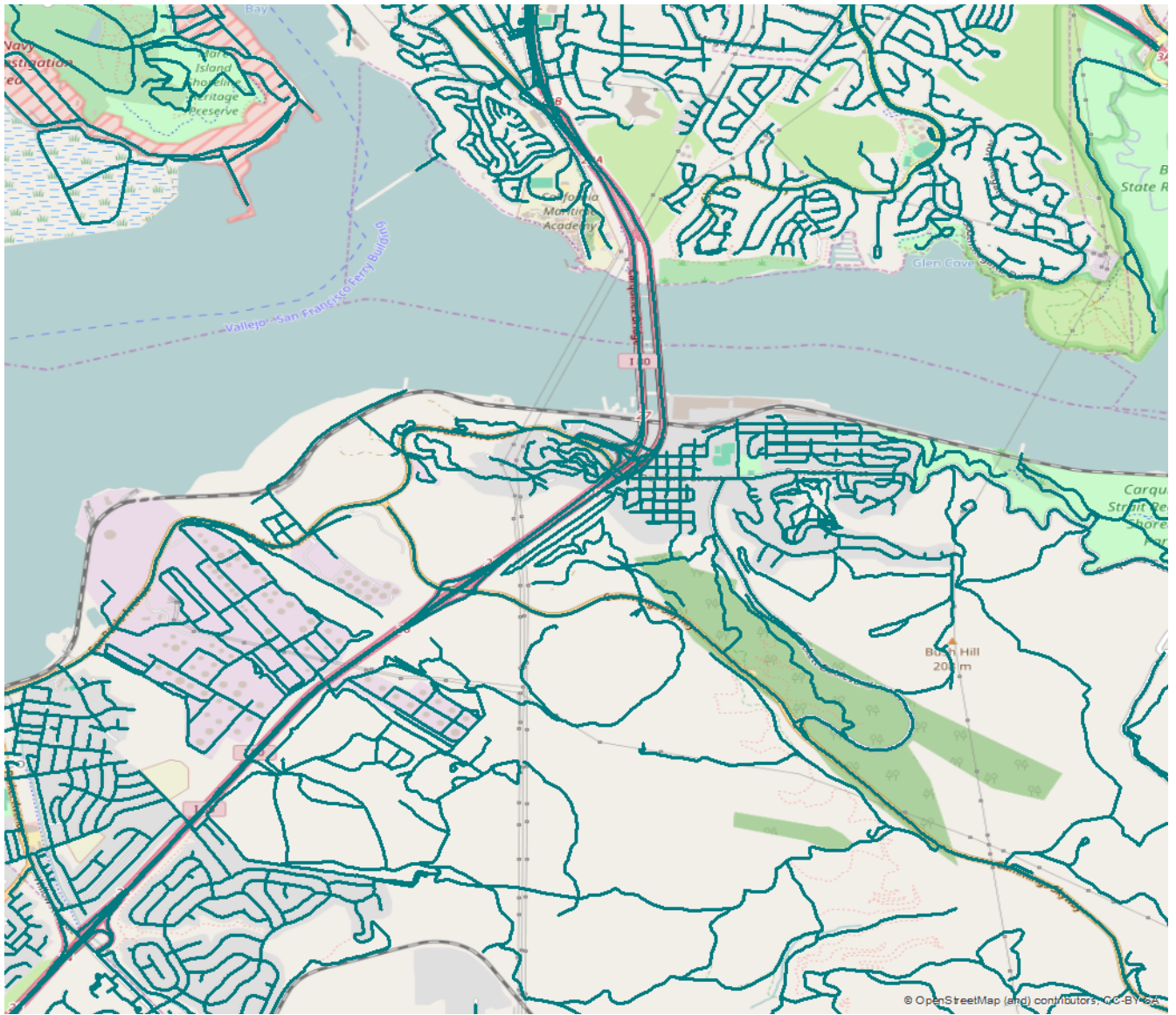


Figure 4. *Trans_RoadSegment* Data in the AOI

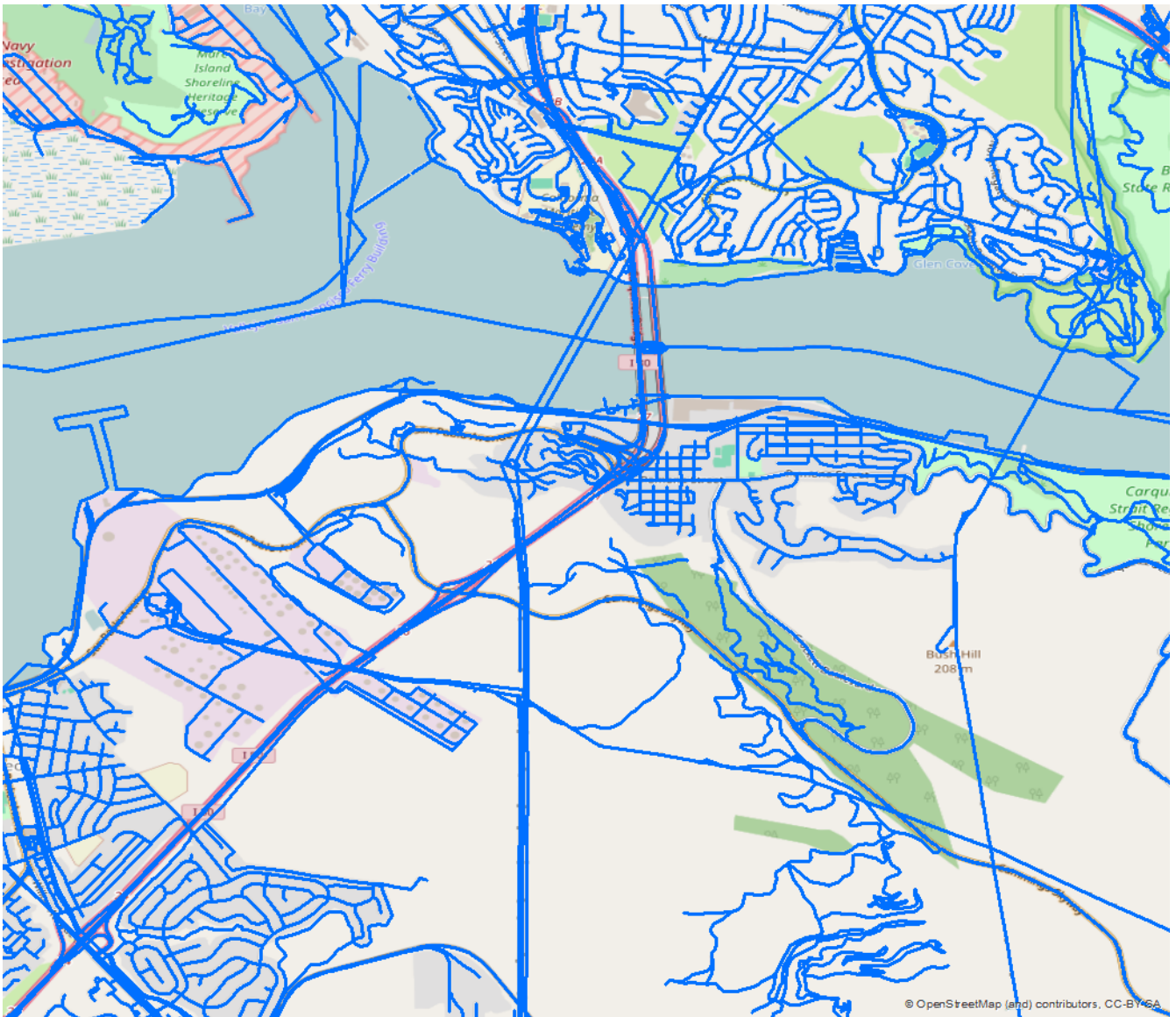


Figure 5. OSM Data in the AOI (as of 9/29/2016)

The Trans_RoadSegment dataset already covers most of the features the OSM data covers. As the OSM data comes directly from the Overpass API [6: <http://overpass-api.de/>], it contains also features like coastlines and power lines.

The Trans_RoadSegment dataset is served by a WFS in the shapefile format. Therefore, a mapping file must be created to map Trans_RoadSegment attributes to OSM attributes. The mapping file is written in Python and looks like the following:

```
#!/bin/python

def translateAttributes(attrs, layerName, geometryType):
    if not attrs: return

    tags = {}

    tags['accuracy'] = '5'

    if 'FULL_STREE' in attrs:
        name = attrs['FULL_STREE']
```

```

if name != 'NULL' and name != '':
    tags['name'] = name

if 'MTFCC_CODE' in attrs:
    mtfcc = attrs['MTFCC_CODE']
    if mtfcc == 'S1100':
        tags['highway'] = 'primary'
    if mtfcc == 'S1200':
        tags['highway'] = 'secondary'
    if mtfcc == 'S1400':
        tags['highway'] = 'unclassified'
    if mtfcc == 'S1500':
        tags['highway'] = 'track'
        tags['surface'] = 'unpaved'
    if mtfcc == 'S1630':
        tags['highway'] = 'road'
    if mtfcc == 'S1640':
        tags['highway'] = 'service'
    if mtfcc == 'S1710':
        tags['highway'] = 'path'
        tags['foot'] = 'designated'
    if mtfcc == 'S1720':
        tags['highway'] = 'steps'
    if mtfcc == 'S1730':
        tags['highway'] = 'service'
    if mtfcc == 'S1750':
        tags['highway'] = 'road'
    if mtfcc == 'S1780':
        tags['highway'] = 'service'
        tags['service'] = 'parking_aisle'
    if mtfcc == 'S1820':
        tags['highway'] = 'path'
        tags['bicycle'] = 'designated'
    if mtfcc == 'S1830':
        tags['highway'] = 'path'
        tags['horse'] = 'designated'

# Is the ISONEWAY attribute properly populated?
if 'ISONEWAY' in attrs and attrs['ISONEWAY'] != '':
    if attrs['FLOW'] == '1':
        tags['oneway'] = 'yes'
    if attrs['FLOW'] == '2':
        tags['oneway'] = 'no'

return tags

```

The name attribute is mapped, as well as the type of the street and an attribute that indicates whether the street is a oneway street.

The result of the conflation is shown in the following image:

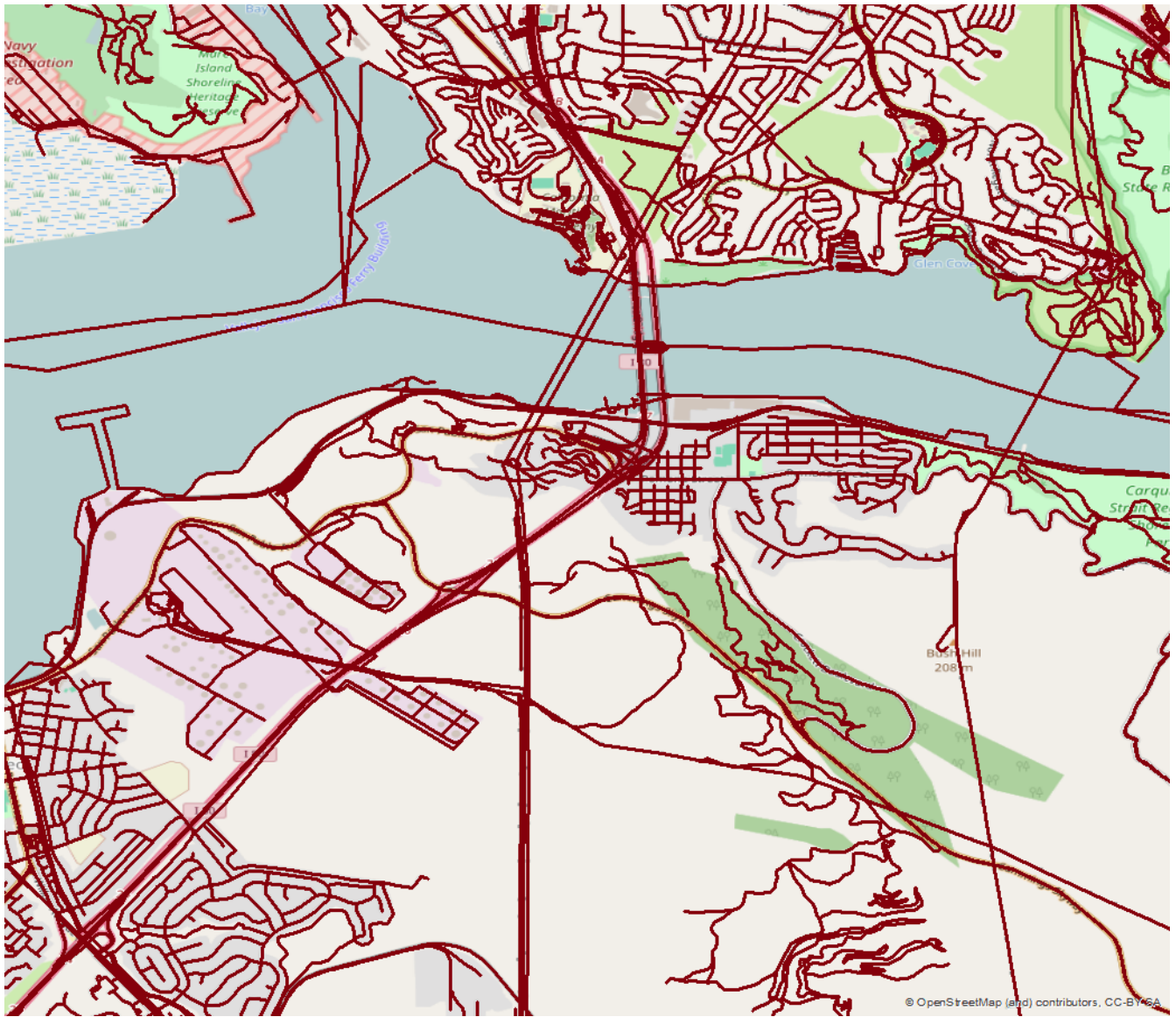


Figure 6. Conflated streets in the AOI

New features include the coastline, power lines and also ferry routes. Most of the OSM road features were used in the conflated dataset. Some features of the Trans_RoadSegment dataset are missing in the conflated result.

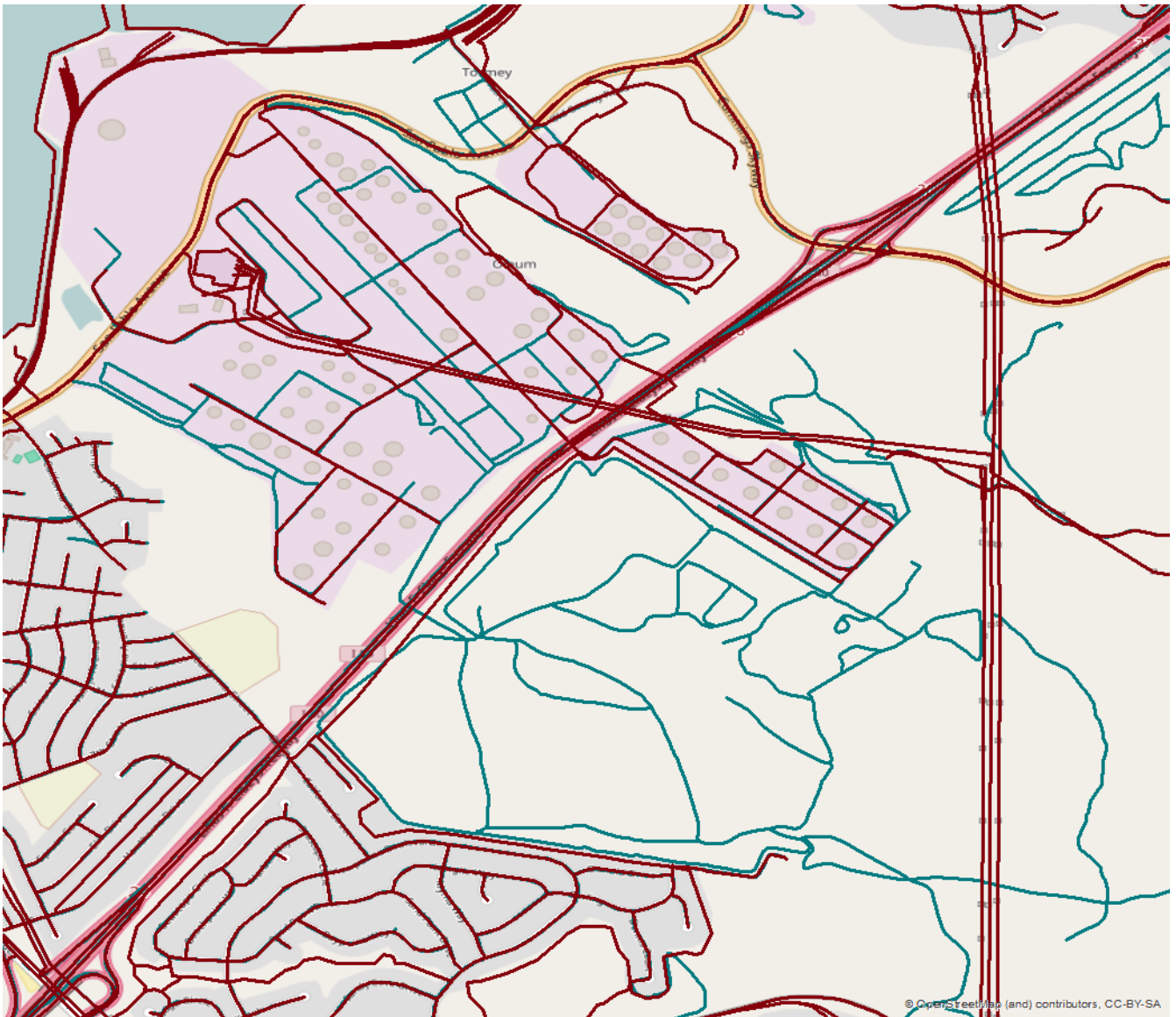


Figure 7. Missing features of the input dataset (Input in green)

The missing features all have the *MTFCC_CODE* attribute set to *S1400*, which is mapped to *highway = unclassified*. The conflated dataset has some positional corrections:

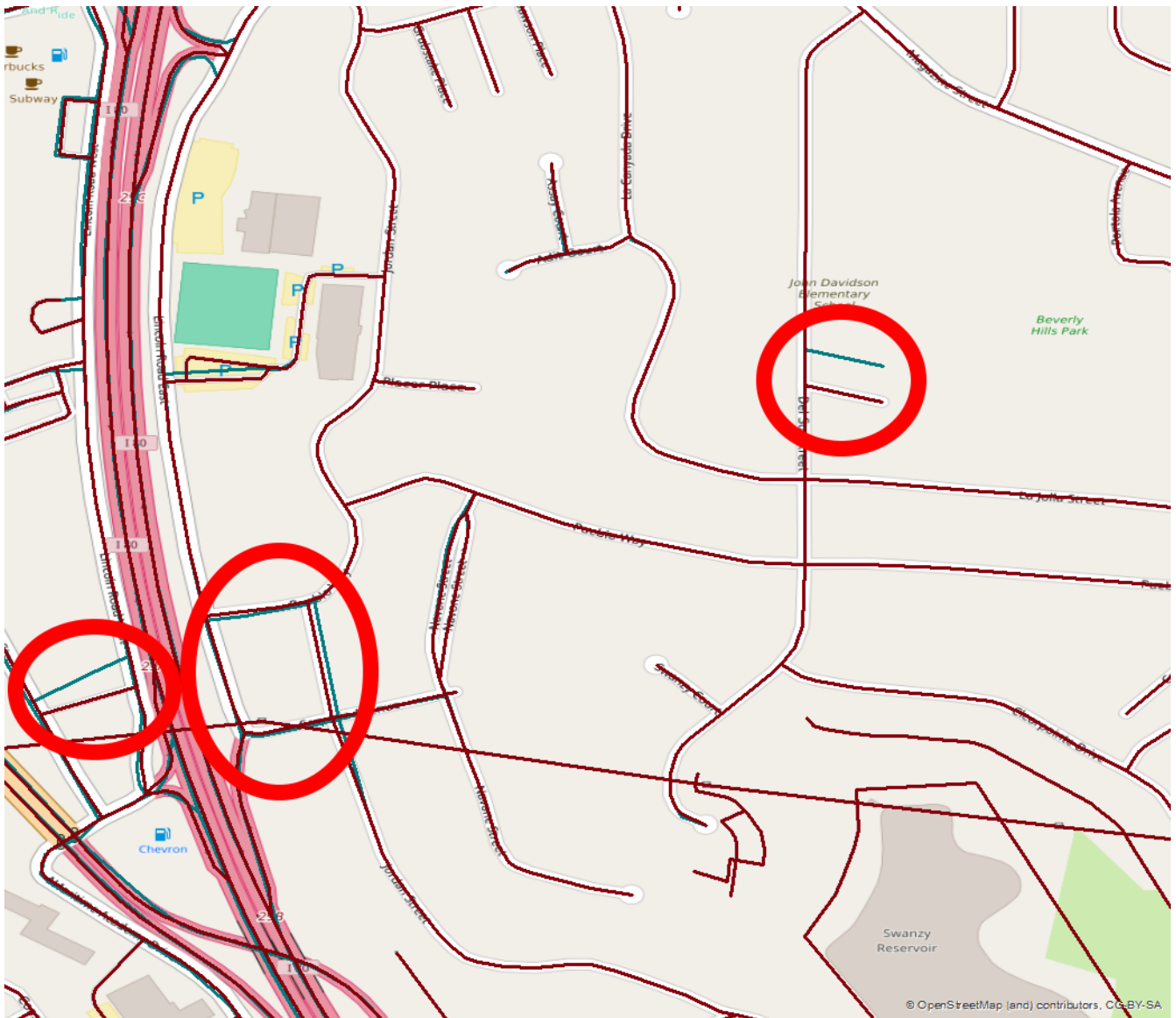


Figure 8. Positional corrections in the conflated dataset (Input in green)

On a technical side, Hootenanny is very well suited for execution in a WPS. The command line interface allows execution of Hootenanny functions from WPS processes written in various languages without effort. The conflation tasks of Testbed-12 were accomplished by Hootenanny without problems. However, the ability to define conflation rules like it is possible using the RoadMatcher software [7: <http://www.vividsolutions.com/products.asp?catg=spaapp&code=roadmatcher>] is not available. The possibility for semi-automated conflation using the Hootenanny WPS process was not investigated and could be a future work item.

Chapter 8. Conflation WPS Profiles

Based on the hierarchical profiling approach defined in the WPS 2.0 standard, we defined different profiles for conflation processes: (1) Conflation process concept (2) Generic conflation process (3) Conflation process implementation profile Profiles can be extended and themselves extend multiple profiles. In the following the profiles for conflation are described.

8.1. Conflation process concept

A process concept describes the functionality of a process or process group on a high level. The WPS 2.0 standard recommends that this could be done using a HTML page. The high level process concept for conflation can be found here: <http://52north.github.io/wps-profileregistry/concept/conflation.html>

8.2. Generic Conflation Process Profile

A generic process profile consists of (1) a XML document similar to a process description but without data formats specified for inputs and outputs and (2) a HTML page with descriptions for the process itself and for the inputs and outputs. The generic conflation process profile can be found here: <http://52north.github.io/wps-profileregistry/generic/conflation.html>

The generic process XML description:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:GenericProcess
  xmlns:ows="http://www.opengis.net/ows/2.0"
  xmlns:wps="http://www.opengis.net/wps/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/2.0
http://schemas.opengis.net/wps/2.0/wps.xsd">

  <ows:Title>Conflation process</ows:Title>
  <ows:Identifier>http://52north.github.io/wps-
profileregistry/generic/conflation.xml</ows:Identifier>
  <ows:Metadata xlink:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/concept" xlink:href="http://52north.github.io/wps-
profileregistry/concept/conflation"/>
  <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://52north.github.io/wps-profileregistry/generic/conflation.html"/>

  <wps:Input>
    <ows:Title>Conflation Input 1</ows:Title>
    <ows:Identifier>INPUT1</ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://52north.github.io/wps-
```

```

profileregistry/generic/conflation.html#input1"/>
  </wps:Input>
  <wps:Input>
    <ows:Title>Conflation Input 2</ows:Title>
    <ows:Identifier>INPUT2</ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://52north.github.io/wps-
profileregistry/generic/conflation.html#input2"/>
  </wps:Input>
  <wps:Input>
    <ows:Title>Reference layer</ows:Title>
    <ows:Identifier><pre>REFERENCE_LAYER</pre></ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://52north.github.io/wps-
profileregistry/generic/conflation.html#reference_layer"/>
  </wps:Input>
  <wps:Output>
    <ows:Title>Conflation output</ows:Title>
    <ows:Identifier>CONFLATION_OUTPUT</ows:Identifier>
    <ows:Metadata
xlink:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlink:href="http://52north.github.io/wps-
profileregistry/generic/conflation.html#conflation_output"/>
  </wps:Output>
</wps:GenericProcess>

```

8.3. Hootenanny Conflation Process Implementation Profile

An implementation profile consists of a WPS process description and a HTML page with details about the implementation and inputs and outputs of the process. The HTML page for the Hootenanny conflation process profile can be found here: <http://52north.github.io/wps-profileregistry/implementing/hootenanny-conflation.html>

The process description looks like the following:

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessOfferings xmlns:wps="http://www.opengis.net/wps/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/2.0
http://schemas.opengis.net/wps/2.0/wps.xsd">
  <wps:ProcessOffering processVersion="1.0.0" jobControlOptions="sync-execute async-
execute" outputTransmission="value reference">
    <wps:Process>

```

```

<ows:Title>Hootenanny Conflation Process</ows:Title>
<ows:Identifier>testbed12.lsa.HootenannyConflation</ows:Identifier>
<ows:Metadata xlin:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/concept" xlin:href="http://52north.github.io/wps-
profileregistry/concept/conflation"/>
  <ows:Metadata xlin:role="http://www.opengis.net/spec/wps/2.0/def/process-
profile/generic" xlin:href="http://52north.github.io/wps-
profileregistry/generic/conflation"/>
  <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation"/>
    <wps:Input minOccurs="1" maxOccurs="1">
      <ows:Title>INPUT1</ows:Title>
      <ows:Abstract>Conflation input 1</ows:Abstract>
      <ows:Identifier>INPUT1</ows:Identifier>
      <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#input1"/>
        <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
          <ns:Format default="true" mimeType="application/x-zipped-shp"/>
          <ns:Format mimeType="application/x-openstreetmap+xml"/>
          <ns:Format mimeType="application/x-zipped-shp" encoding="base64"/>
          <ns:Format mimeType="application/x-zipped-gdb"/>
          <ns:Format mimeType="application/x-zipped-gdb" encoding="base64"/>
        </wps:ComplexData>
      </wps:Input>
      <wps:Input minOccurs="0" maxOccurs="1">
        <ows:Title>INPUT1_TRANSLATION</ows:Title>
        <ows:Abstract>Translation file for conflation input 1</ows:Abstract>
        <ows:Identifier>INPUT1_TRANSLATION</ows:Identifier>
        <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#input1_translation"/>
          <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
            <ns:Format default="true" mimeType="text/x-script.python"/>
            <ns:Format mimeType="text/plain"/>
          </wps:ComplexData>
        </wps:Input>
        <wps:Input minOccurs="1" maxOccurs="1">
          <ows:Title>INPUT2</ows:Title>
          <ows:Abstract>Conflation input 2</ows:Abstract>
          <ows:Identifier>INPUT2</ows:Identifier>
          <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#input2"/>
            <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
              <ns:Format default="true" mimeType="application/x-openstreetmap+xml"/>

```

```

<ns:Format mimeType="text/xml"/>
<ns:Format mimeType="application/x-zipped-shp"/>
<ns:Format mimeType="application/x-zipped-shp" encoding="base64"/>
<ns:Format mimeType="application/x-zipped-gdb"/>
<ns:Format mimeType="application/x-zipped-gdb" encoding="base64"/>
</wps:ComplexData>
</wps:Input>
<wps:Input minOccurs="0" maxOccurs="1">
  <ows:Title>CONFLATION_TYPE</ows:Title>
  <ows:Identifier>CONFLATION_TYPE</ows:Identifier>
  <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#conflation_type"/>
    <wps:LiteralData xmlns:ns="http://www.opengis.net/wps/2.0">
      <ns:Format default="true" mimeType="text/plain"/>
      <ns:Format mimeType="text/xml"/>
      <LiteralDataDomain>
        <ows:AllowedValues>
          <ows:Value>average</ows:Value>
          <ows:Value>reference</ows:Value>
        </ows:AllowedValues>
        <ows:DataType ows:reference="xs:string"/>
      </LiteralDataDomain>
    </wps:LiteralData>
  </wps:Input>
  <wps:Input minOccurs="0" maxOccurs="1">
    <ows:Title>MATCH_THRESHOLD</ows:Title>
    <ows:Abstract>The threshold for calling a relationship a match. Defaults to
0.6. The higher the value the lower the TPR, but likely also the lower the
FPR.</ows:Abstract>
    <ows:Identifier>MATCH_THRESHOLD</ows:Identifier>
    <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#miss_threshold"/>
      <wps:LiteralData xmlns:ns="http://www.opengis.net/wps/2.0">
        <ns:Format default="true" mimeType="text/plain"/>
        <ns:Format mimeType="text/xml"/>
        <LiteralDataDomain>
          <ows:AnyValue/>
          <ows:DataType ows:reference="xs:double"/>
          <ows:DefaultValue>0.6</ows:DefaultValue>
        </LiteralDataDomain>
      </wps:LiteralData>
    </wps:Input>
    <wps:Input minOccurs="0" maxOccurs="1">
      <ows:Title>MISS_THRESHOLD</ows:Title>
      <ows:Abstract>The threshold for calling a relationship a miss. Defaults to
0.6. The higher the value the lower the TNR, but likely also the lower the
FNR.</ows:Abstract>

```

```

<ows:Identifier>MISS_THRESHOLD</ows:Identifier>
<ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#match_threshold"/>
  <wps:LiteralData xmlns:ns="http://www.opengis.net/wps/2.0">
    <ns:Format default="true" mimeType="text/plain"/>
    <ns:Format mimeType="text/xml"/>
    <LiteralDataDomain>
      <ows:AnyValue/>
      <ows:DataType ows:reference="xs:double"/>
      <ows:DefaultValue>0.6</ows:DefaultValue>
    </LiteralDataDomain>
  </wps:LiteralData>
</wps:Input>
<wps:Input minOccurs="0" maxOccurs="1">
  <ows:Title>REFERENCE_LAYER</ows:Title>
  <ows:Abstract>The reference layer which will be dominant tags. Default is 1
and if 2 selected, layer 2 tags will be dominant with layer 1 as geometry snap
layer.</ows:Abstract>
  <ows:Identifier>REFERENCE_LAYER</ows:Identifier>
  <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#reference_layer"/>
    <wps:LiteralData xmlns:ns="http://www.opengis.net/wps/2.0">
      <ns:Format default="true" mimeType="text/plain"/>
      <ns:Format mimeType="text/xml"/>
      <LiteralDataDomain>
        <ows:AllowedValues>
          <ows:Value>1</ows:Value>
          <ows:Value>2</ows:Value>
        </ows:AllowedValues>
        <ows:DataType ows:reference="xs:integer"/>
        <ows:DefaultValue>1</ows:DefaultValue>
      </LiteralDataDomain>
    </wps:LiteralData>
  </wps:Input>
<wps:Output>
  <ows:Title>CONFLATION_OUTPUT</ows:Title>
  <ows:Identifier>CONFLATION_OUTPUT</ows:Identifier>
  <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#conflation_output"/>
    <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
      <ns:Format default="true" mimeType="application/x-zipped-shp"/>
      <ns:Format mimeType="application/x-zipped-shp" encoding="base64"/>
      <ns:Format mimeType="application/x-zipped-gdb"/>
      <ns:Format mimeType="text/xml"
schema="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"/>

```

```

        <ns:Format mimeType="application/vnd.geo+json"/>
    </wps:ComplexData>
</wps:Output>
<wps:Output>
    <ows:Title>CONFLATION_REPORT</ows:Title>
    <ows:Identifier>CONFLATION_REPORT</ows:Identifier>
    <ows:Metadata
xlin:role="http://www.opengis.net/spec/wps/2.0/def/process/description/documentation"
xlin:href="http://52north.github.io/wps-profileregistry/implementing/hootenanny-
conflation#conflation_report"/>
        <wps:ComplexData xmlns:ns="http://www.opengis.net/wps/2.0">
            <ns:Format default="true" mimeType="text/plain"/>
        </wps:ComplexData>
    </wps:Output>
</wps:Process>
</wps:ProcessOffering>
</wps:ProcessOfferings>

```

The implementation profile extends the generic conflation profile and additionally defines the input and output formats. As a blue-print the process description for a WPS 1.0.0 process that is shipped with the Hootenanny webapp (see section 6.2) was used. This could establish interoperability, if a new future version of Hootenanny supports WPS 2.0. Also, it will ease the migration process for the build-in Hootenanny WPS.

Chapter 9. Recommendations

The profiles developed in Testbed-12 belong to the first WPS 2.0 profiles ever created. They serve as proof-of-concept and blueprints for future profiling efforts. However, the clients used in Testbed-12 did not support profiling, i.e. they could not handle the metadata information regarding the profiles. This reduces the usefulness of the profiles that should foster interoperability between clients and servers developed by different vendors. For future testbed initiatives, clients should be able to understand profile metadata and possibly servers from different vendors, implementing the same profile, should be developed. Then the usefulness of the profiles can be tested accordingly. Two additional recommendations for future work:

- Conflation workflow including revision of conflation results: The current conflation workflow is based on automated conflation, i.e. there is no user interaction during a conflation run, possibly leading to a lower quality result. This could be enhanced by allowing the user to interact during a conflation run, e.g. by checking (parts) of the result and run the conflation with modified parameters.
- Use algorithms to compute data quality: During testbed-12, a WPS offering algorithms to compute data quality was developed. Combining the conflation WPS and the data quality WPS was out of scope for Testbed-12, but could be worth investigating in subsequent Testbed initiatives. Cascading WPS execute request are part of the WPS specification. Also, the profiling approach could help with a possible automated combination of conflation and data quality WPS.

Appendix A: Hootenanny Conflation Log

1.1. JSON Execute Request

The log below was created after sending the following JSON request to the endpoint:

<http://tb12.dev.52north.org/wps-rest/processes/testbed12.lsa.HootenannyConflation/jobs/>

```
{
  "Execute": {
    "Identifier": "testbed12.lsa.HootenannyConflation",
    "Input": [
      {
        "Reference": {
          "_mimeType": "application/x-zipped-shp",
          "_href":
"http://geoprocessing.demo.52north.org:8080/data/Trans_RoadSegment-aoi.zip"
        },
        "_id": "INPUT1"
      },
      {
        "Reference": {
          "_mimeType": "text/x-script.python",
          "_href":
"http://geoprocessing.demo.52north.org:8080/data/TNM_Roads.py"
        },
        "_id": "INPUT1_TRANSLATION"
      }
    ]
  }
}
```



```

    {
      "Reference": {
        "_mimeType": "application/x-openstreetmap+xml",
        "_href":
"http://geoprocessing.demo.52north.org:8080/data/sf_only_roads-aoi.osm"
      },
      "_id": "INPUT2"
    }
  ],
  "output": [{
    "_mimeType": "application/x-zipped-shp",
    "_id": "CONFLATION_OUTPUT",
    "_transmission": "reference"
  }, {
    "_mimeType": "text/plain",
    "_id": "CONFLATION_REPORT",
    "_transmission": "reference"
  }],
  "_service": "WPS",
  "_version": "2.0.0"
}
}

```

1.2. Log

This log was created in the asciidoc format by Hootenanny, it was modified to fit in this document.

- Schema: TDSv61.js

Conflation Type

- Type:

Conflation Match Thresholds

- Building: 0.6
- Highway: 0.161
- POI: 0.6

Conflation Miss Thresholds

- Building: 0.6
- Highway: 0.999
- POI: 0.6

Conflation Review Thresholds

- Building: 0.6
- Highway: 0.25
- POI: 0.6

Conflation Search Radii

- Highway: -1
- All Other Features: -1

Notes

- In the Hootenanny command replace any occurrences of '!semi!' with ';' before testing.

1.3. Summary

A summary of the most important statistics for the conflation job just executed are presented in the bar graph and table below. Shown are the total and conflatable number of features for both inputs, the total number of features for the output, and a breakdown of the output features by those that are conflated, marked for review, and unmatched.

Conflatable features are defined as those that are supported by Hootenanny's matching algorithms. Generally speaking, Hootenanny support types of highways, buildings and POIs. Other features are not supported by Hootenanny. Note also that conflatables is defined by the current settings of the matching algorithm. The matching algorithm can be configured by setting different "match creators", where each supports different types of features. By default, a full set of match creators is applied, but this could be reduced to a subset of feature types depending on the user's desired product needs.

The total number of features in the output have no correlation with the two inputs' numbers of features. There are a few reasons for this: (1) features from the inputs are split in order to perform matching, (2) features are added during specific conflation algorithm steps, and (3) the merge step of the conflation may change the number of features.

The output features are classified by the following breakdown: (1) conflated - those features that successfully combined from multiple sources and possibly from the same input; (2) unmatched - those features where no matches were found between any other, and possibly from the same source; and (3) reviewable - those features with rankings falling between the conflated and unmatched classifications where manual review by a user is required.

The bar graph shows the number of features for each category, and the table provides the actual count numbers and percentages where appropriate. The percentage of conflatables are relative to the total number of features for the respective input. The percentages for the conflated, reviewable, and unmatched are relative to the total number of features in the output. Note for now we do not show percentage values of how many features were conflated from the inputs to the output because there is no easy way to calculate and record these values while considering the feature splits and the impact on the performance of Hootenanny's conflation.

```
unset key

set style data histogram

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0
```

```

set xtic rotate by -70 scale 0 font "arial,10"
set ytics rotate by 0 offset -1,0,0
set key vertical outside
set yrange [0:2964]
set title "Summary of Important Statistics"
set ylabel "Number of Features"
unset xtics
set xtic rotate by -70 center offset 1.3,-3.3 scale 0 font "arial,10"
set nokey
set size 1,1
set terminal png size 600,450
set bmargin at screen 0.3
plot '-' using 2:xtic(1) ti col, '' u 2 ti col
unknownKey
"Input 1 Total Features"      703
"Input 2 Total Features"     1036
"Input 1 Conflatables  "    696
"Input 2 Conflatables  "    1036
"Output Total Features  "    2470
"Conflated Features   "     435
"Reviewable Features   "     33
"Unmatched Features   "    2035
e

```

Table 5. Conflation Summary Chart

	Count	Percentage
Input 1 Total Features	703	NA
Input 2 Total Features	1036	NA
Input 1 Conflatables	696	99%
Input 2 Conflatables	1036	100%
Output Total Features	2470	NA
Conflated Features	435	17.6%
Reviewable Features	33	1.3%
Unmatched Features	2035	82.4%

1.4. Statistical Aspects of the Input and Output Datasets

Statistics for a variety of aspects of the input and output datasets are presented in this section.

1.4.1. Summary of Basic Feature Elements

There are three basic elements of OpenStreetMap's conceptual data model of the physical world. These are nodes (points in space), ways (linear and area features), and relations (a high-level construct that helps coordinate how ways and nodes work together). To go along with these are tags which can associate with any of the three basic elements. In this section, a summary of the number of each element and tags are shown in the bar graph. For each, a comparison of the number of features for the two inputs and output are shown.

Note: More information about the OSM Elements is provided at: <http://wiki.openstreetmap.org/wiki/Elements>

```
reset

unset y2label

set title "Summary of Basic Feature Elements"

set ylabel "Number of Features"

unset key

set style data histogram

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0

set xtic rotate by -70 scale 0 font "arial,10"

set ytics rotate by 0 offset -1,0,0

set key vertical outside

set yrange [0:27414]
```

```

plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col

"Input 1"

"Nodes" 15573

"Ways" 2233

"Relations" 0

"Tags" 3486

e

"Input 2"

"Nodes" 11239

"Ways" 1036

"Relations" 0

"Tags" 10211

e

"Output"

"Nodes" 14817

"Ways" 2498

"Relations" 50

"Tags" 22845

e

```

Table 6. Feature Count Histogram

	Input 1	Input 2	Output
Nodes	15573	11239	14817
Ways	2233	1036	2498
Relations	0	0	50
Tags	3486	10211	22845

1.4.2. Summary of Input Features by Type

Some of the key feature types that Hootenanny operates upon are 'buildings', 'highways', and 'points of interest (POIs)'. These types are defined using the naming conventions that OSM assigns for these attributes. The buildings type are the area footprints for what the title indicates—buildings. The highways type is a bit more generic and refers to a collection of linear segments that include road types (highways, streets, dirt roads), cart tracks, and trails. The POIs are defined for Hootenanny as nodes with a attribute "poi=yes". The POIs are typically derived from an OSM attribute like "amenity". For example, POIs may be a "place of worship", "cafe", etc.

A bar graph and table are presented below showing information about the input datasets. The bar graph shows the total features, number of conflatable features, and a breakdown of the features by types for each of the two inputs. The feature types presented are: buildings, highways, and POIs. The number of conflatables is a count of features that Hootenanny supports (see definition described previously). The table shows the specific count values, percentage of conflatables relative to the total features, and percentage of feature type relative to the number of conflatables.

```
unset key

set style data histogram

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0

set xtic rotate by -70 scale 0 font "arial,10"

set ytics rotate by 0 offset -1,0,0

set key vertical outside

set yrange [0:1243.2]

set title "Breakdown of Features by Type for Inputs 1 & 2"

set ylabel "Number of Features"

unset xtics

set xtic rotate by -70 center offset 0.0,-2 scale 0 font "arial,10"

set size 1,1
```



```

set nokey

set terminal png size 600,450

set bmargin at screen 0.3

plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col

"Input 1"

"Total Features"      703

"Conflatables  "      696

"Buildings    "       0

"Highways     "      696

"POIs         "       0

e

"Input 2"

"Total Features"     1036

"Conflatables  "     1036

"Buildings    "       0

"Highways     "     1036

"POIs         "       0

e

```

Table 7. Input Features by Type

	Input 1	Input 2
Total Features	703	1036
Conflatables	696 (99%)	1036 (100%)
Buildings	0 (0%)	0 (0%)
Highways	696 (100%)	1036 (100%)
POIs	0 (0%)	0 (0%)

1.4.3. Summary of Output Features by Type

A breakdown of the features by type for the output are described in this section. The bar graph shows statistics for all features, and the breakdown by types of buildings, highways, and POIs. The statistics shown are the total number of features, number conflated, number marked for review, and number of unmatched features. The table shows the specific count values and the percentages of the conflated, marked for review, and unmatched relative to the total counts of the respective feature type.

```
unset key

set style data histogram

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0

set xtic rotate by -70 scale 0 font "arial,10"

set ytics rotate by 0 offset -1,0,0

set key vertical outside

set yrange [0:2964]

set title "Breakdown of Features by Type for the Output"

set ylabel "Number of Features"

unset xtics

set xtic rotate by -70 center offset 0.5,-2 scale 0 font "arial,10"

set size 1,1

set key

set terminal png size 600,450

set bmargin at screen 0.3

plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col, '' u 2 ti col
```

```
"Total Count"
"All Features" 2470
"Buildings"    " 0
"Highways"    " 2463
"POIs"        " 0
```

e

```
"Conflated"
"All Features" 435
"Buildings"    " 0
"Highways"    " 435
"POIs"        " 0
```

e

```
"Marked Review"
"All Features" 33
"Buildings"    " 0
"Highways"    " 33
"POIs"        " 0
```

e

```
"Unmatched"
"All Features" 2035
"Buildings"    " 0
"Highways"    " 2028
"POIs"        " 0
```

e

Table 8. Output Features by Type

	Total Count	Conflated	Marked Review	Unmatched
All Features	2470	435 (17.6%)	33 (1.3%)	2035 (82.4%)
Buildings	0	0 (0%)	0 (0%)	0 (0%)
Highways	2463	435 (17.7%)	33 (1.3%)	2028 (82.3%)
POIs	0	0 (0%)	0 (0%)	0 (0%)

1.4.4. Summary of Area Features

Area features are the collection of OSM attributes that encompass a geographic area like a building footprint. Hootenanny supports conflation of building type area features to date. In this section, area measurements are presented for the total area features, the building features, and the output conflated buildings for both the inputs and output (where appropriate). The area measurements are calculated in square kilometers and presented in the bar graph and tables.

The area measurements provide an estimate for how well Hootenanny performed conflation on the input data. If the two input datasets do not overlap geographically, the output area should be the sum of the two input areas and the conflated output area equal to 0. This also implies the output area should never be greater than the sum of the input areas.

Another rule to understand is it is correct to assume that the output area is always greater than or equal to the input with the largest area. However, the results presented in the report here take into consideration that some features are marked for manual review (i.e. those not marked automatically as matches nor misses for conflation). These cases are not included in the area measurement calculations for the output because their eventual outcome is unknown until the user makes a decision for each. This brings up a good point for future work to decide if the features marked for review should be included in the area measurements, even though it is unknown what the exact output area will be before the decision, an estimate could be made that approximates the logical rule defined here.

```
reset
unset y2label
set title "Summary of Area Features"
set ylabel "Kilometers Squared"
unset key
set style data histogram
set style histogram cluster gap 1
set style fill solid 0.5 border -1
set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"
set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"
set boxwidth 0.8
set size 1.0,1.0
set xtic rotate by -70 scale 0 font "arial,10"
set ytics rotate by 0 offset -1,0,0
```

```

set key vertical outside
set xrange [0:1]
plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col
"Input 1"
"Area Features" 0
"Buildings" 0
"Conflated Buildings" 0
e
"Input 2"
"Area Features" 0
"Buildings" 0
"Conflated Buildings" 0
e
"Output"
"Area Features" 0
"Buildings" 0
"Conflated Buildings" 0
e

```

Table 9. Output Area Histogram

	Input 1	Input 2	Output
Area Features	0	0	0
Buildings	0	0	0
Conflated Buildings	NA	NA	0

1.4.5. Summary of Linear Features

Linear features are the collection of OSM attributes that have a path-like structure and do not make up an area. Hootenanny supports the highways type (described in an earlier section) to date. In this section, length measurements are presented for the total linear features, the highway type features, and the output conflated highways. The length measurements are calculated in kilometers and presented in the bar graph and tables.

The length measurements provide an estimate for how well Hootenanny performed conflation on the input data. Like the area features, if the two input datasets do not overlap geographically, the output length should be the sum of the two input lengths and the conflated output length equal to 0. This also implies the output length should never be greater than the sum of the input lengths.

The next rule to understand is similar to the area feature's second rule. It is correct to assume that the output length is always greater than or equal to the input with the longest length. However, the results presented in the report here take into consideration that some features are marked for manual review (i.e., those not marked automatically as matches nor misses for conflation). Like for area measurements, these cases are also not included in the length measurement calculations for the output because their eventual outcome is unknown until the user makes a decision for each. The idea of estimating the lengths associated with the features marked for review will be discussed at a future point.

```
reset

unset y2label

set title "Summary of Linear Features"

set ylabel "Kilometers"

unset key

set style data histogram

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0

set xtic rotate by -70 scale 0 font "arial,10"

set ytics rotate by 0 offset -1,0,0
```

```

set key vertical outside

set xrange [0:455.984]

plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col

"Input 1"

"Linear Features"    120.921

"Highways"          120.921

"Conflated Highways"  0

e

"Input 2"

"Linear Features"    297.327

"Highways"          330.772

"Conflated Highways"  0

e

"Output"

"Linear Features"    346.542

"Highways"          379.986

"Conflated Highways"  54.1991

e

```

Table 10. Linear Features

	Input 1	Input 2	Output
Linear Features	120.921	297.327	346.542
Highways	120.921	330.772	379.986
Conflated Highways	NA	NA	54.1991

1.4.6. Summary of POI Features

Points of Interest (POIs) are defined as nodes in Hootenanny, and are typically derived from the OSM attribute "amenity", which may contain features like: "place of worship", "cafe", etc. The statistics presented in this section show the total number of POIs for the two inputs and the output, and the number of conflated POIs in the output. The data is presented in the bar graph and table provided.

The POI counts for the output should never be greater than the sum of the counts for the two inputs. The maximum count for the output is reached when all of the POIs from both inputs are not matches for conflation. This implies that any two POIs do not overlay geographically within a specified radius and their names are not similar. If these conditions are completely satisfied, the output POI count reaches a maximum by summing the POIs from both inputs.

It is also intuitive to think that the output count should always be greater than or equal to the maximum input count. But this is not correct if there exists any matching POIs from within the same input. A match within an input is referred to as a "self conflation" when the PLACES POI conflation algorithm is being used. Note, a match between inputs is known as a "challenge conflation" for the PLACES algorithm.

There is also another case where the output may be lower than the expected and that occurs when POIs are combined due to their close spatial proximity (e.g., two Starbucks locations located within a half mile may get merged into one if the POI radius of influence is too large). This may occur within and/or between the two inputs. This issue is documented in the Hootenanny Algorithms manual section "The Starbucks Problem".

```
reset

unset y2label

set title "Summary of POI Features"

set ylabel "Number of Features"

unset key

set style data histogram

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0
```

```

set xtic rotate by -70 scale 0 font "arial,10"

set ytics rotate by 0 offset -1,0,0

set key vertical outside

set yrange [0:1]

plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col

"Input 1"

"POIs" 0

"Conflated POIs" 0

e

"Input 2"

"POIs" 0

"Conflated POIs" 0

e

"Output"

"POIs" 0

"Conflated POIs" 0

e

```

Table 11. POI Features

	Input 1	Input 2	Output
POIs	0	0	0
Conflated POIs	NA	NA	0

1.4.7. Summary of Unique Names

A summary of the unique names used for the features are shown in the bar graph in this section. The number of unique names for the two inputs and the output are shown in the left-most bar. A breakdown of the unique names by the types of building and highways are shown in the right-most bars. The table shows the specific counts for each data source and type.

```
reset

unset y2label

set title "Breakdown of Unique Names by Type"

set ylabel "Number of Names"

unset key

set style data histogram

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0

set xtic rotate by -70 scale 0 font "arial,10"

set ytics rotate by 0 offset -1,0,0

set key vertical outside

set yrange [0:907.2]

plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col

"Input 1"

"All Names" 372

"Building Names"    0

"Highway Names" 369

e
```

"Input 2"

"All Names" 401

"Building Names" 0

"Highway Names" 401

e

"Output"

"All Names" 756

"Building Names" 0

"Highway Names" 753

e

Table 12. Unique Names

	Input 1	Input 2	Output
All Names	372	401	756
Building Names	0	0	0
Highway Names	369	401	753

1.4.8. Summary of Tags

Tags describe specific attributes of the OSM data elements 'nodes', 'ways', and 'relations' (defined in an earlier section). The tags in Hootenanny are classified in two categories: information and metadata. Metadata tags contain data about the feature's provenance (for example, who created it, where it came from, UUID, etc.). Information tags contain general information associated with the features (basically all the rest of the attributes). The bar graph shows the total number of tags and a breakdown of the number of information and metadata tags for the two inputs and output datasets. The table shows the specific counts for each category.

```
reset

unset y2label

set title "Summary of Tags"

set ylabel "Number of Tags"

unset key

set style data histogram

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0

set xtic rotate by -70 scale 0 font "arial,10"

set ytics rotate by 0 offset -1,0,0

set key vertical outside

set yrange [0:27414]

plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col

"Input 1"

"All Tags" 3486

"Information Tags" 1253
```

"Metadata Tags" 2233

e

"Input 2"

"All Tags" 10211

"Information Tags" 9151

"Metadata Tags" 1060

e

"Output"

"All Tags" 22845

"Information Tags" 20323

"Metadata Tags" 2522

e

Table 13. Summary of Tags

	Input 1	Input 2	Output
All Tags	3486	10211	22845
Information Tags	1253	9151	20323
Metadata Tags	2233	1060	2522

1.4.9. Summary of Translated Tags

Translated tags are generated in Hootenanny by applying the translator to the inputs and conflated output data. It applies the translation schema defined in Section 2 and then collects information about what was generated to build the statistics for the translated tags. There are three types of translated tags: populated, default, and null.

- Populated tags are tags that have been assigned non-default values. In other words, values populated from the source by the translation script.
- Default tags are tags that have been assigned the default value from the translation schema. For example, from the translation schema TDSv61:

```
{ name:"CAA",  
  desc:"Controlling Authority",  
  optional:"R",  
  defValue:"-999999",  
  type:"enumeration",  
  <...>  
}
```

If the translated output used the default value 'CAA == "-999999"' then the tag is a translated default tag.

- Null tags are tags that have been assigned a null value to the output where no default value (from the schema) nor any translated value was available.

The number of each translated tag type for the inputs and the conflated output are shown in the bar graph and table. The breakdown of each tag type (default, populated, and null) are presented with the specific values provided in the table.

```
reset  
unset y2label  
set title "Summary of Translated Tags"  
set ylabel "Number of Tags"  
unset key  
set style data histogram
```

```

set style histogram cluster gap 1

set style fill solid 0.5 border -1

set grid ytics lt 0 lw 1 lc rgb "#bbbbbb"

set grid xtics lt 0 lw 1 lc rgb "#bbbbbb"

set boxwidth 0.8

set size 1.0,1.0

set xtic rotate by -70 scale 0 font "arial,10"

set ytics rotate by 0 offset -1,0,0

set key vertical outside

set yrange [0:225878]

plot '-' using 2:xtic(1) ti col, '' u 2 ti col, '' u 2 ti col

"Input 1"

"Translated Populated Tags" 4014

"Translated Default Tags" 53754

"Translated Null Tags" 51504

e

"Input 2"

"Translated Populated Tags" 5917

"Translated Default Tags" 76048

"Translated Null Tags" 80602

e

"Output"

"Translated Populated Tags" 14244

"Translated Default Tags" 184130

"Translated Null Tags" 188232

e

```


Table 14. Translated Tags

	Input 1	Input 2	Output
Translated Populated Tags	4014	5917	14244
Translated Default Tags	53754	76048	184130
Translated Null Tags	51504	80602	188232

1.5. Hootenanny Run-time Performance

A summary of the performance results by time for each of the processing steps of the conflation are shown in the bar graph. The processing components measured are: read inputs, stats for inputs 1 and 2, stats for output, apply named operations, apply preprocessing operations, projections to planar, find matches, optimize matches, create mergers, apply mergers, apply post operations, projections to WGS84, write output, old road conflation, overall conflation, and other times. The bar chart shows measurements in seconds and the table (on the next page) with precise timing in minutes, seconds, and milliseconds and percentage of overall time. The table also shows some aggregate timing measurements: old road conflation (if applicable), conflation (aggregate of the core processing steps), and total (aggregate of all the steps). The category 'other' is the residual timing information.

```
X=(0,5.2,0,0.2,0,0.2,0.3,0,0.6,4.1,0,0,0.8,6,2.8,2,0.3)
```

```
val=X
```

```
pos=arange(17)+.5
```

```
figure(1)
```

```
barh(pos,val,align='center')
```

```
yticks(pos, ('Other','Conflation','Old Road Conflation', 'Write Output', 'Project to  
WGS84', 'Apply Post Ops', 'Apply Mergers', 'Create Mergers', 'Optimize Matches', 'Find  
Matches', 'Project to Planar', 'Apply Pre Ops', 'Apply Named Ops', 'Stats for Output',  
'Stats for Input 2', 'Stats for Input 1', 'Read Inputs'))
```

```
xlabel('Time (s)')
```

```
title('Timing of Processing Steps')
```

Table 15. Timing of Processing Steps

Processing Step	Time
Read Inputs	0m 0s 315ms (1.8%)
Stats for Input 1	0m 2s 18ms (11.7%)
Stats for Input 2	0m 2s 806ms (16.2%)
Stats for Output	0m 5s 971ms (34.5%)
Apply Named Ops	0m 0s 754ms (4.4%)
Apply Pre Ops	0m 0s 0ms (0%)
Project to Planar	0m 0s 0ms (0%)
Find Matches	0m 4s 84ms (23.6%)
Optimize Matches	0m 0s 646ms (3.7%)
Create Mergers	0m 0s 5ms (0%)
Apply Mergers	0m 0s 315ms (1.8%)
Apply Post Ops	0m 0s 177ms (1%)
Project to WGS84	0m 0s 16ms (0.1%)
Write Output	0m 0s 216ms (1.2%)
Old Road Conflation	Not applicable
Conflation (aggregate of several steps)	0m 5s 227ms (30.2%)
Other	0m 0s 0ms (0%)
Total	0m 17s 324ms

Appendix B: Revision History

Table 16. Revision History

Date	Release	Editor	Primary clauses modified	Descriptions
April 12, 2016	B. Pross	.1	all	initial version
April 15, 2016	B. Pross	.1	all	Added outline and relevance section
April 15, 2016	B. Pross	.1	all	Added outline and relevance section
August 17, 2016	B. Pross	.1	7	Added road conflation example
September 29, 2016	B. Pross	.1	all	Updated content
September 30, 2016	B. Pross	.1	all	Updated content
October 10, 2016	B. Pross	.1	6	Updated background section
October 11, 2016	B. Pross	.1	6	Updated background section
October 14, 2016	B. Pross	.1	all	Incorporated comments of review

Appendix C: Bibliography

- [1] Rieke, Matthes and Benjamin Pross,: OGC® OWS-9 Cross Community Interoperability (CCI) Conflation with Provenance Engineering Report. (2013).
- [2] Masó, Joan et al.: OGC® Testbed 10 Provenance Engineering Report (2014)
- [3] Longley, Paul A. et al.: Geographic Information Systems and Science. Wiley and Sons. (2001)
- [4] Jeffe, Mike J. et al.: Hootenanny User Guide v0.2.26 (2016)