

Open Geospatial Consortium

Approval Date: 2013-01-18

Posted Date: 2013-02-05

Reference number of this document: OGC 12-159

Reference URL for this document: www.opengeospatial.net/def/doc-type/per/cci-conflation-provenance

Category: Public Engineering Report

Editors: Matthes Rieke; Benjamin Pross

OGC[®] OWS-9 Cross Community Interoperability (CCI) Conflation with Provenance Engineering Report

Copyright © 2013 Open Geospatial Consortium.

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is 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.

Document type: OGC[®] Engineering Report
Document subtype: NA
Document stage: Approved for public release
Document language: English

Abstract

This OGC[®] Engineering Report describes the architecture of a WPS capable of conflating two datasets while capturing provenance information about the process. The report also provides information about defining and encoding conflation rules and about encoding provenance information.

This Engineering Report was created as a deliverable for the OGC Web Services, Phase 9 (OWS-9) initiative of the OGC Interoperability Program.

Keywords

ogcdoc, ows9, conflation, provenance, wps, sparql, wfs

What is OGC Web Services 9 (OWS-9)?

OWS-9 builds on the outcomes of prior OGC interoperability initiatives and is organized around the following threads:

- **Aviation:** Develop and demonstrate the use of the Aeronautical Information Exchange Model (AIXM) and the Weather Exchange Model (WXXM) in an OGC Web Services environment, focusing on support for several Single European Sky ATM Research (SESAR) project requirements as well as FAA (US Federal Aviation Administration) Aeronautical Information Management (AIM) and Aircraft Access to SWIM (System Wide Information Management) (AAtS) requirements.
- **Cross-Community Interoperability (CCI):** Build on the CCI work accomplished in OWS-8 by increasing interoperability within communities sharing geospatial data, focusing on semantic mediation, query results delivery, data provenance and quality and Single Point of Entry Global Gazetteer.
- **Security and Services Interoperability (SSI):** Investigate 5 main activities: Security Management, OGC Geography Markup Language (GML) Encoding Standard Application Schema UGAS (UML to GML Application Schema) Updates, Web Services Façade, Reference Architecture Profiling, and Bulk Data Transfer.
- **OWS Innovations:** Explore topics that represent either new areas of work for the Consortium (such as GPS and Mobile Applications), a desire for new approaches to existing technologies to solve new challenges (such as the OGC Web Coverage Service (WCS) work), or some combination of the two.
- **Compliance & Interoperability Testing & Evaluation (CITE):** Develop a suite of compliance test scripts for testing and validation of products with interfaces

implementing the following OGC standards: Web Map Service (WMS) 1.3 Interface Standard, Web Feature Service (WFS) 2.0 Interface Standard, Geography Markup Language (GML) 3.2.1 Encoding Standard, OWS Context 1.0 (candidate encoding standard), Sensor Web Enablement (SWE) standards, Web Coverage Service for Earth Observation (WCS-EO) 1.0 Interface Standard, and TEAM (Test, Evaluation, And Measurement) Engine Capabilities.

The OWS-9 sponsors are: AGC (Army Geospatial Center, US Army Corps of Engineers), CREAM-GeoViQua-EC, EUROCONTROL, FAA (US Federal Aviation Administration), GeoConnections - Natural Resources Canada, Lockheed Martin Corporation, NASA (US National Aeronautics and Space Administration), NGA (US National Geospatial-Intelligence Agency), USGS (US Geological Survey), UK DSTL (UK MoD Defence Science and Technology Laboratory).

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.

Contents		Page
1	Introduction.....	1
1.1	Scope	1
1.2	Document contributor contact points	1
1.3	Revision history.....	1
1.4	Future work	1
1.5	Foreword	2
2	References.....	2
3	Terms and definitions	3
4	Conventions	3
4.1	Abbreviated terms	3
4.2	UML notation	4
4.3	Used parts of other documents	4
5	OWS-9 CCI Conflation with Provenance Engineering Report	5
5.1	CCI Architecture	5
5.2	Data for Conflation.....	5
5.2.1	USGS Data.....	6
5.2.2	NGA Data	6
5.3	Use Cases for Conflation.....	6
5.3.1	Use Case 1 – Add Features and Attributes	6
5.3.2	Use Case 2 – Add and Update Feature and Attributes.....	6
5.3.3	Use Case 3 – Quality Update Feature and Attributes	6
5.3.4	Architecture.....	7
5.4	Conflation WPS Profiles	9
5.4.1	Entry Point WPS	9
5.4.1.1	Inputs	11
5.4.1.2	Outputs	12
5.4.1.3	Workflow.....	12
5.4.2	Geometry Conflation WPS	14
5.4.2.1	Inputs	17
5.4.2.2	Outputs	17
5.4.2.3	Workflow.....	18
5.5	Conflation Rules.....	18
5.5.1	Geometrical Conflation.....	19
5.5.1.1	Encoding.....	19
5.5.2	Attribute Conflation	21
5.6	Provenance	22
5.6.1	Conflation provenance.....	22
5.6.1.1	Identity and Entity Properties.....	22
5.6.1.2	Data Lineage Properties	22

5.6.2	Encoding	23
5.6.2.1	Conflation provenance encoding at dataset level	24
5.6.2.2	Conflation provenance encoding at feature level	24
5.6.3	Provenance Realization within OWS-9	25
6	Lessons learned and open issues	27
7	Accomplishments.....	27
8	Recommendations.....	27

Figures	Page
Figure 1 – CCI Architecture Overview	5
Figure 2 – Conflation WPS Architecture Overview.....	7
Figure 3 – General Conflation Workflow.	9
Figure 4 – High-level Geometry Conflation Workflow.....	18
Figure 5 - Exemplary Visualization of Provenance.....	28

Tables	Page
Figure 1 – CCI Architecture Overview	5
Figure 2 – Conflation WPS Architecture Overview.....	7
Figure 3 – General Conflation Workflow.	9
Figure 4 – High-level Geometry Conflation Workflow.....	18
Figure 5 - Exemplary Visualization of Provenance.....	28

OGC® OWS-9 CCI Conflation with Provenance Engineering Report

1 Introduction

1.1 Scope

This OGC® Engineering Report describes the architecture of a WPS capable of conflating two datasets while capturing provenance information about the process. It also provides information about defining and encoding conflation rules and about encoding provenance information.

1.2 Document contributor contact points

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

Name	Organization
Matthes Rieke (MR)	52°North
Benjamin Pross (BPR)	52°North
Genong Yu (GY)	George Mason University
Matt Sorenson (MS)	National Geospatial-Intelligence Agency

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
30.05.2012	0.01	BPR	All	Initial Document
30.11.2012	0.5	MR	All	Enhanced all clauses to reflect the current state of the OWS-9 Conflation Architecture.
21.12.2012	1.0	MR	Lessons learned, future work, recommendations	Preparation of the final ER

1.4 Future work

- Support for multiple source datasets** – The current architecture only supports the conflation from one source to one target dataset. An interesting

feature for the enhancement of the Conflation architecture would be the support for multiple source datasets which would be conflated to one target dataset. Additional sources could include but are not limited to VGI data or Twitter feeds.

- **Explicit service chaining** – The architecture developed in OWS-9 makes use of two WPS instances for the overall conflation. Geometry and attribute conflation are separated into two different processes. Though, they are both invoked from one aggregation process in a non-interoperable way. Future work could focus on the design of a process chains (e.g. Client -> Geometry Process -> Attribute Process -> Client) for improved transparency. Such work would include decisions on interim data formats for both the dataset and the provenance tracking.
- **Enhancement of provenance information** – The current architecture does not benefit from the capabilities of ISO 19115 Metadata. Future work should cover the encoding of provenance and quality of data through ISO 19115 and the integration of such data into Catalogue services. Another interesting work area within metadata and provenance is the visualization of sources and process steps. See section 8 for some examples.
- **Extending the ProcessDefinitions** – The current process definitions are limited regarding the configuration of the underlying conflation rules. This has been decided with respect to the complexity of the overall task and the convenient integration into the client user interface. For future work it would be very interesting to add configuration parameters to reflect all conflation rules. This would allow an improved assessment of the results on the client side by comparing different configurations.

1.5 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.

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 11-063, OWS-8 CCI Semantic Mediation ER

OGC 06-121r3, OGC® Web Services Common Standard

OGC 05-007r7, OpenGIS® Web Processing Service

OGC Revision Notes for OpenGIS® Implementation Specification: Geographic information - Geography Markup Language Version 3.2.1

NOTE This OWS Common Standard contains a list of normative references that are also applicable to this Implementation Standard.

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r3] 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.

3.2 WPS profile

According to the WPS specification (OGC 05-007r7), a WPS Application Profile should be defined as a standalone document containing:

- An Universal Resource Name (URN) that uniquely identifies the process A reference response to a DescribeProcess request for that process (reference process schema),
- A human-readable description of the process and its implementation (optional, but recommended).
- A Web Service Description Language (WSDL) document for that process (optional).

4 Conventions

4.1 Abbreviated terms

OGC	Open Geospatial Consortium
OWS-9	OGC Web Services Initiative, Phase 9
TDS	Topographic Data Store

TNM	The National Map
WFS	Web Feature Service
WFS-G	Web Feature Service Gazetteer
WFS-T	WFS-Transactional
WMS	Web Map Service
WPS	Web Processing Service

4.2 UML notation

Some diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3]. Additionally, UML sequence diagrams are used to illustrate workflow.

4.3 Used parts of other documents

This document uses significant parts of document [OGC 07-160 r1]. To reduce the need to refer to that document, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are shown with a light grey background (15%).

5 OWS-9 CCI Conflation with Provenance Engineering Report

This document summarizes the work carried out within OWS-9 in the field of dataset conflation and Web Processing Service. It provides details on the general service architecture, service interfaces as well as on internal workflows developed within this testbed.

5.1 CCI Architecture

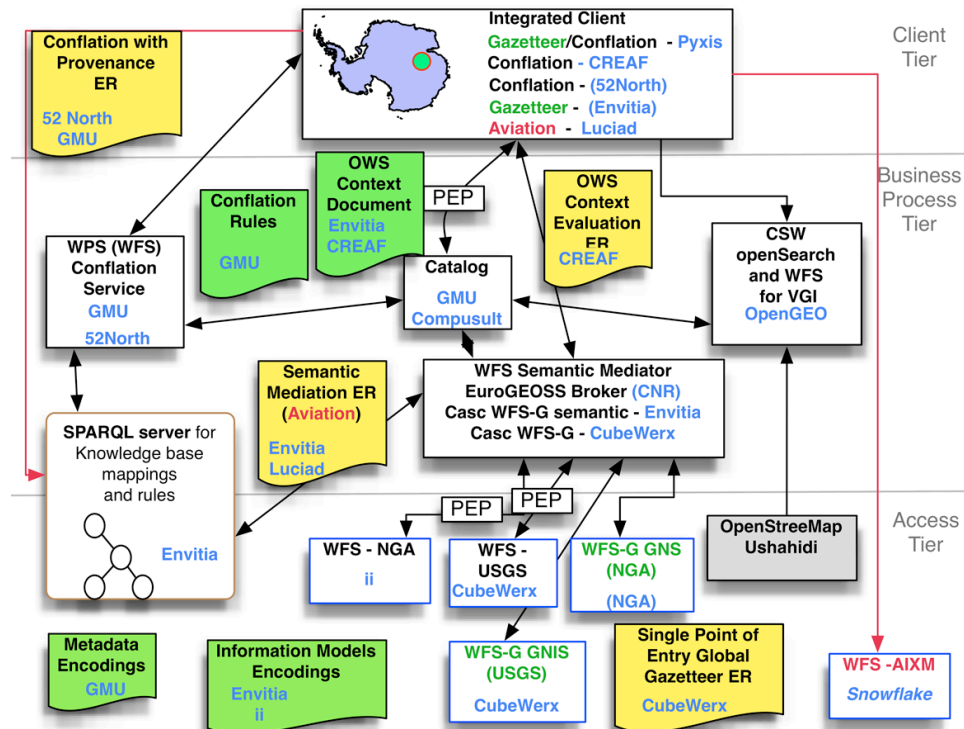


Figure 1 – CCI Architecture Overview

Figure 1 shows the overall OWS-9 CCI Architecture. The conflation service is located in the business tier having relations to several services in the access tier (SPARQL server, WFS's).

5.2 Data for Conflation

Dataset conflation comes into play when data from two different sources needs to be aligned and integrated. The scenario applied in this testbed uses data from “The National Map” (TNM, administrated by USGS) and the “Topographic Data Store” (TDS, provided by NGA).

The starting point for conflation is the assumption that the currently available data in TDS is not entirely complete and thus needs an update by another dataset. Here, the TNM data is used to enrich the TDS data with additional features and attributes.

5.2.1 USGS Data

The TNM data used within this testbed has a high feature count and is provided through a WFS version 1.1 or 2.0. It can be obtained as GML version 3.1 or 3.2. It was used as the “source” dataset within OWS-9.

5.2.2 NGA Data

The TDS data has less features compared to the TNM data. It was also provided through a dedicated WFS (version 1.1 or 2.0) as GML version 3.2. Within OWS-9 it was used as the “target” dataset for conflation.

5.3 Use Cases for Conflation

Several Use Cases have been defined in prior to the design of the conflation architecture. They range from geometric conflation of point features to polygonal features.

5.3.1 Use Case 1 – Add Features and Attributes

The aim of this use case is to add missing features and attributes from USGS TNM (source dataset) to NGA TDS (target dataset). Background: The user holds a fairly complete dataset (NGA, Monterey) and wants to identify items in alternate dataset (USGS, Monterey) and only add information that is not present in the existing dataset.

5.3.2 Use Case 2 – Add and Update Feature and Attributes

An outdated target (reference) dataset is updated with a more recent source dataset. For matched geometry features and attributes use the source and include attributes from the source that do not exist in the reference. Integrate unmatched geometry features and attributes from the source into the reference dataset.

5.3.3 Use Case 3 – Quality Update Feature and Attributes

Employ rigorous matching algorithms to identify “matched” geometric features between the target (reference) and source datasets. A new conflated product will be developed (shapefile) that combines the matched features from the reference dataset with the unmatched features from the target and source datasets with specific geometries and attributes selected based on conflation rules. Application of conflation rules may be fully automated or require analyst review for ambiguous or critical features. The result will be an enhanced product that eliminates duplicate geometry but retains the attribution, as well as improves the spatial coverage.

5.3.4 Architecture

This section is intended to describe the CCI Conflation Architecture which is mainly based on the chaining of WPS servers that consume data provided by WFS instances. An outline of the developed architecture is shown in Figure 2. Additionally, Figure 3 illustrates the high-level workflow via a sequence diagram.

The main entry point for the user is a WPS client which allows defining the input data sets as well as starting the execution of the conflation process. For this purpose, the conflation WPS server accepts two WFS references that provide the according inputs. One of the WFS references points to the so called target WFS server which contains the dataset that shall be updated. The other WFS server allows accessing the source data set which contains the features and attributes that shall be used for updating the target data set.

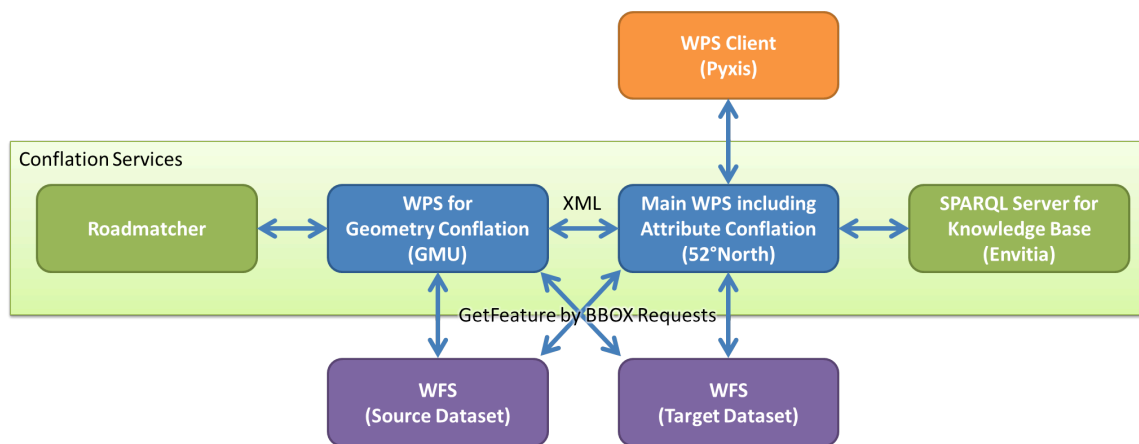


Figure 2 – Conflation WPS Architecture Overview

The conflation process consists of two separate tasks:

- Geometry Conflation: Determining if a feature of the source data set is contained in the target data set; if it is contained, both features are linked, if it is not contained, the feature from the source data sets is added to the target data set
- Attribute Conflation: Both, the source and the target data set may possess different attributes; during the attribute conflation semantic reasoning is applied to determine equivalencies between attributes and if an attribute contained in the source data set is missing in the target data set, it is added to the target data set.

For each of these conflation steps a separate WPS server has been implemented:

- WPS for Geometry Conflation developed by GMU
- WPS for Attribute Conflation developed by 52°North

As a first step, the Geometry Conflation WPS is invoked by the main WPS process. For performing the Geometry Conflation, the GMU WPS relies on the RoadMatcher plugin for the OpenJUMP GIS¹. The results are returned to the main WPS as XML instance of an on-the-fly schema (including provenance information).

Subsequently, the 52°North WPS executes the Attribute Conflation. For this purpose, the Envia SPARQL Server/Knowledge Base is used on top of the Geometry Conflation result. After that, all features (conflated geometries and attributes) are transformed back into the target GML schema.

Provenance information using the gmd:MD_Metadata element is created on a dataset and feature level. To create this metadata, both the geometry provenance information and the attribute conflation processes, are taken into account.

Finally, both results of the conflation process are returned to the client.

¹ http://sourceforge.net/projects/jump-pilot/files/OpenJUMP_plugins/More%20Plugins/Roadmatcher%201.4%20for%20OJ/

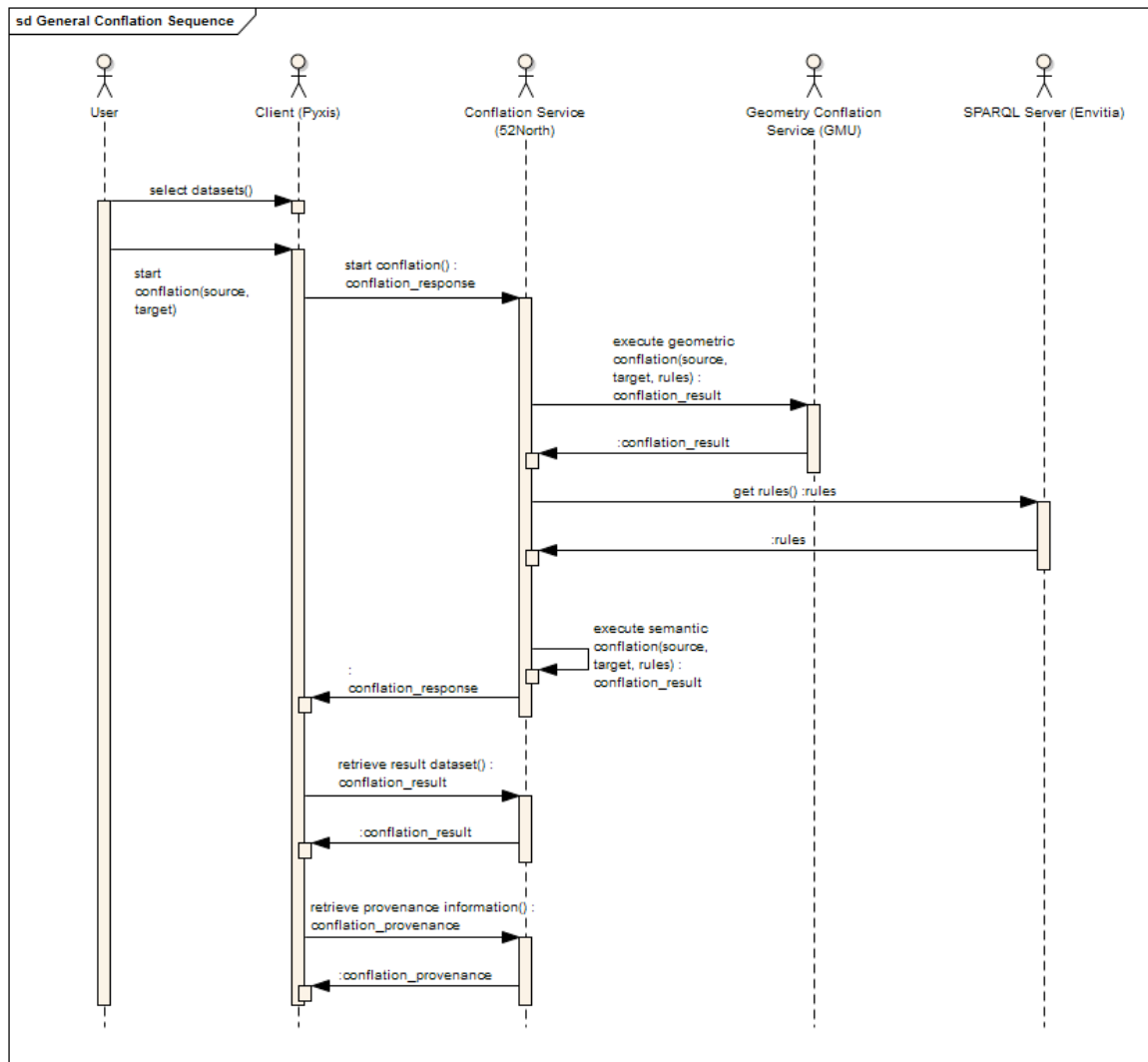


Figure 3 – General Conflation Workflow.

5.4 Conflation WPS Profiles

The following sections describe the two developed WPS Profiles – the general entry point Profile for the client software and the internally invoked geometry conflation Profile.

5.4.1 Entry Point WPS

As the geometry and attribute conflation is combined in a single WPS process a client uses this WPS process as the entry point for invoking the dataset conflation. Internally, the geometry conflation WPS process is invoked and attribute conflation applied on top of the result.

Listing 1 provides the main WPS ProcessDescription used by the client to setup the ExecuteRequest.

Listing 1 – Entry Point WPS ProcessDescription.

```

<wps:ProcessDescriptions xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd" xml:lang="en-US"
service="WPS" version="1.0.0">
  <ProcessDescription xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
wps:processVersion="2" storeSupported="true" statusSupported="false">
    <ows:Identifier>org.n52.wps.conflation.ConflationProcess</ows:Identifier>
    <ows:Title>ConflationProcess</ows:Title>
    <ows:Abstract>Process for conflating two different datasources located at Monterey,
CA, US</ows:Abstract>
    <ows:Metadata xlink:title="conflation"/>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>source-wfs</ows:Identifier>
        <ows:Title>Source WFS instance</ows:Title>
        <ows:Abstract>Source WFS instance as GetFeature request</ows:Abstract>
        <ComplexData>
          <Default>
            <Format>
              <MimeType>text/xml; subtype=gml/3.2.1</MimeType>
            </Format>
          </Default>
          <Supported>
            <Format>
              <MimeType>text/xml; subtype=gml/3.2.1</MimeType>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>target-wfs</ows:Identifier>
        <ows:Title>Target WFS instance</ows:Title>
        <ows:Abstract>Target WFS instance as GetFeature request</ows:Abstract>
        <ComplexData>
          <Default>
            <Format>
              <MimeType>text/xml; subtype=gml/3.2.1</MimeType>
            </Format>
          </Default>
          <Supported>
            <Format>
              <MimeType>text/xml; subtype=gml/3.2.1</MimeType>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
      <Input minOccurs="0" maxOccurs="1">
        <ows:Identifier>bbox</ows:Identifier>
        <ows:Title>BBOX</ows:Title>
        <ows:Abstract>Area of Interest for conflation, axis order as defined by the
CRS.
If not provided, a default value for Monterey, CA, US will be used.
</ows:Abstract>
        <BoundingBoxData>
          <Default>
            <CRS>urn:ogc:def:crs:EPSG::4326</CRS>
          </Default>
        </BoundingBoxData>
      </Input>
    </DataInputs>
  </ProcessDescription>
</wps:ProcessDescriptions>

```



```

        </Default>
        <Supported>
          <CRS>urn:ogc:def:crs:EPSG::4326</CRS>
        </Supported>
      </BoundingBoxData>
    </Input>
  </DataInputs>
  <ProcessOutputs>
    <Output>
      <ows:Identifier>result</ows:Identifier>
      <ows:Title>Conflation results</ows:Title>
      <ows:Abstract>Conflation results</ows:Abstract>
      <ComplexOutput>
        <Default>
          <Format>
            <MimeType>text/xml; subtype=gml/3.2.1</MimeType>
          </Format>
        </Default>
      </ComplexOutput>
      <Schema>http://schemas.opengis.net/gml/3.2.1/feature.xsd</Schema>
    </Output>
    <Output>
      <ows:Identifier>provenance</ows:Identifier>
      <ows:Title>Provenance information</ows:Title>
      <ows:Abstract>Provenance information about the conflation
process</ows:Abstract>
      <ComplexOutput>
        <Default>
          <Format>
            <MimeType>text/xml; subtype=gml/3.2.1</MimeType>
          </Format>
        </Default>
      </ComplexOutput>
      <Schema>http://schemas.opengis.net/gml/3.2.1/feature.xsd</Schema>
    </Output>
    <Output>
      <ows:Identifier>provenance</ows:Identifier>
      <ows:Title>Provenance information</ows:Title>
      <ows:Abstract>Provenance information about the conflation
process</ows:Abstract>
      <ComplexOutput>
        <Default>
          <Format>
            <MimeType>text/xml; subtype=gml/3.2.1</MimeType>
          </Format>
        </Default>
      </ComplexOutput>
      <Schema>http://schemas.opengis.net/gml/3.2.1/feature.xsd</Schema>
    </Output>
  </ProcessOutputs>
</ProcessDescription>
</wps:ProcessDescriptions>

```

5.4.1.1 Inputs

The profile defines three inputs.

- source-wfs** – a WFS GetFeature request providing the source dataset
- target-wfs** – a WFS GetFeature request providing the target dataset
- bbox** – an optional bounding box where the conflation should be applied.

If no bounding box is provided, the process sets this parameter to a predefined default value (the Monterey area). The process takes both WFS GetFeature requests as the basis

and adds the bounding box specifics to the URL. An example **target-wfs** value looks like the following:

```
http://services.interactive-instruments.de/xsprojects/ows8-tds/cgi-bin/1tds/wfs?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&NAMESPACE=xm1ns(tds=ht tp:%2F%2Fmetadata.dod.mil%2Fmdr%2Fns%2FGSIP%2F3.0%2Ftds%2F3.0)&TYPENAME=tds:Road Geocurve&OUTPUTFORMAT=text/xml;%20subtype=gml/3.2.1.
```

After adding the bounding box specifics it results in the following request:

```
http://services.interactive-instruments.de/xsprojects/ows8-tds/cgi-bin/1tds/wfs?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&NAMESPACE=xm1ns(tds=ht tp:%2F%2Fmetadata.dod.mil%2Fmdr%2Fns%2FGSIP%2F3.0%2Ftds%2F3.0)&TYPENAME=tds:Road Geocurve&OUTPUTFORMAT=text/xml;%20subtype=gml/3.2.1&SRSNAME=urn:ogc:def:crs:EPSG ::4326&BBOX=36.625,-121.795,36.65,-121.695,urn:ogc:def:crs:EPSG::4326.
```

The adjusted request is used for the invocation of the geometry conflation WPS process.

5.4.1.2 Outputs

After applying both the geometry and attribute conflation the data is transformed back into the target dataset schema. It is provided to the user within the wps:Output with identifier “**result**”.

Besides the resulting dataset the process provides provenance information on the resulting dataset through the wps:Output with identifier “**provenance**”. The details of this output are described in section 5.6.

5.4.1.3 Workflow

In a first step the process takes the input WFS GetFeature URLs and includes the bounding box specifics as described in section 5.4.1.1. The newly defined GetFeature URLs are then used within a WPS Execute request to the Geometry Conflation WPS (see following sections).

In parallel, the process prepares itself for the attribute conflation. First, a query to the Sparql KB server to retrieve all mappings (see [OGC 11-063] for details) defined from the source to the target schema is invoked. An exemplary query is provided in Listing 2. In a second step, all attributes of the source and target schemas are resolved and stored internally to allow the post-processing of the Geometry Conflation WPS result.

Listing 2 – Sparql Query for Retrieval of Mappings.

```
DESCRIBE ?a
  WHERE {
    ?a <http://env032011.appspot.com/ows9.rdf#sourceFeatureType>
      <http://www.usgs.gov/projects/ows8#RoadSegmentGeocurve> .
    ?a <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://env032011.appspot.com/ows9.rdf#Mapping> .
    ?a <http://env032011.appspot.com/ows9.rdf#targetFeatureType>
      <http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#RoadGeocurve>
  }
```

Once the result of the Geometry Conflation WPS is available, the attributes of each feature are mapped back against the previously resolved attributes of the source and target datasets.

Listing 3 – Excerpt of a Geometry Conflation Response.

```

<ogr:Resultout fid="Resultout.0">
  <ogr:geometryProperty><gml:LineString
srsName="EPSG:4326"><gml:coordinates>36.612322705,-121.843961511999979
36.61231368599999,-121.842956293000015 36.61226066499999,-
121.840209803999954</gml:coordinates></gml:LineString></ogr:geometryProperty>
  <ogr:SOURCE>f1</ogr:SOURCE>
  <ogr:MAXDIST>6.0837689036198705</ogr:MAXDIST>
  <ogr:ADJSIZE>0.0000000000000000</ogr:ADJSIZE>
  <ogr:TRIMDIST>4.0461997504864833</ogr:TRIMDIST>
  <ogr:NEARNESS>1.0000000000000000</ogr:NEARNESS>
  <ogr:ADJUSTED>N</ogr:ADJUSTED>
  <ogr:ADJANGDEL>0.0000000000000000</ogr:ADJANGDEL>
  <ogr:SRCSTATE>Matched (Reference)</ogr:SRCSTATE>
  <ogr:RESSTATE>Inconsistent</ogr:RESSTATE>
  <ogr:SPLITSTART>N</ogr:SPLITSTART>
  <ogr:SPLITEND>N</ogr:SPLITEND>
  <ogr:MANMATCH>N</ogr:MANMATCH>
  <ogr:MANADJ>N</ogr:MANADJ>
  <ogr:MANADJCNT>0</ogr:MANADJCNT>
  <ogr:MATCHORIEN>Y</ogr:MATCHORIEN>
  <ogr:REVIEWED>N</ogr:REVIEWED>
  <ogr:UPDATETIME>2012-11-30 09:07:45</ogr:UPDATETIME>
  ...
  <ogr:oneWay>noInformation</ogr:oneWay>
  <ogr:thoroughfareClass>noInformation</ogr:thoroughfareClass>
  <ogr:thoroughfareType>street</ogr:thoroughfareType>
  ...
  <ogr:gml_id>CWFID.TR_ROADS.0.13977</ogr:gml_id>
  <ogr:permanentIdentifier>{A3003F40-A42F-4481-B556-CC4C8BC121A7}
  </ogr:permanentIdentifier>
  <ogr:roadClass>10003</ogr:roadClass>
  <ogr:isOneWay>0</ogr:isOneWay>
  ...
</ogr:Resultout>

```

Listing 3 provides an exemplary excerpt of a Geometry Conflation WPS response. The red-colored attributes are automatically generated parameters of the RoadMatcher tool which are used as quality indicators within the provenance generation.

The green-shaded attributes are derived from the target and the yellow-shaded from the source dataset. As both source and target attributes are qualified by the same namespace the service needs to resolve their origin by comparing its local names with those defined in the original schemas. This approach has certain drawback which is discussed in section **Error! Reference source not found.**

Once the result attributes have been assigned to source and target dataset the attribute conflation can be applied. Here, the attribute conflation rules (see section 5.5.2) are applied. The process defaults currently to the “Add-only” use case (“identify attributes from source not present in target and add in target with target schema”) – a future work item is the dynamic setup of the attribute conflation behavior by the client software.

As most of the attributes in the source dataset are defined as values of a static enumerations (e.g. “isOneWay” can only have “0” or “1” as a value), an additional mapping from the source attribute enumeration value to the target attribute enumeration

value needs to be resolved from the Sparql server. An exemplary query is provided in Listing 4. These value-level mappings are the used to transform the source attribute values into the target values resulting in a conflation result which is valid against the target XML schema.

Listing 4 – Sparql Query for Retrieval of Value Enumeration Mappings.

```
DESCRIBE <http://www.usgs.gov/projects/ows8#RoadIsOnewayType_yes>
```

5.4.2 Geometry Conflation WPS

The geometry conflation is executed by a dedicated WPS instance. Listing 5 defines its ProcessDescription.

Listing 5 – Geometry Conflation WPS ProcessDescription.

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:ProcessDescriptions xmlns:wfs="http://www.opengis.net/wps/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd" xml:lang="en-US"
service="WPS" version="1.0.0">
  <ProcessDescription statusSupported="true" storeSupported="true"
wfs:processVersion="1.0.0">
    <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
edu.gmu.csiss.conflation.wps.RoadMatcherConflation</ows:Identifier>
    <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
edu.gmu.csiss.conflation.wps.RoadMatcherConflation</ows:Title>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier
xmlns:ows="http://www.opengis.net/ows/1.1">inputGMLURL1</ows:Identifier>
        <ows:Title
xmlns:ows="http://www.opengis.net/ows/1.1">inputGMLURL1</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier
xmlns:ows="http://www.opengis.net/ows/1.1">inputGMLversion1</ows:Identifier>
        <ows:Title
xmlns:ows="http://www.opengis.net/ows/1.1">inputGMLversion1</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier
xmlns:ows="http://www.opengis.net/ows/1.1">inputGMLURL2</ows:Identifier>
        <ows:Title
xmlns:ows="http://www.opengis.net/ows/1.1">inputGMLURL2</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier
xmlns:ows="http://www.opengis.net/ows/1.1">inputGMLversion2</ows:Identifier>
        <ows:Title
xmlns:ows="http://www.opengis.net/ows/1.1">inputGMLversion2</ows:Title>
```

```

        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchMinimumLineSegmentLength</ows:Identifier>
        <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchMinimumLineSegmentLength</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:double"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchNearnessTolerance</ows:Identifier>
        <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchNearnessTolerance</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:double"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchFindStandaloneRoadEnabled</ows:Identifier>
        <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchFindStandaloneRoadEnabled</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchFindStandaloneRoadMinimumDistance</ows:Identifier>
        <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchFindStandaloneRoadMinimumDistance</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:double"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchFindMatchedRoadsEnabled</ows:Identifier>
        <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchFindMatchedRoadsEnabled</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchFindMatchedRoadMaximumDistance</ows:Identifier>
        <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoMatchFindMatchedRoadMaximumDistance</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:double"/>
          <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoAdjustDatasetName</ows:Identifier>
        <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
          AutoAdjustDatasetName</ows:Title>
        <LiteralData>
          <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"

```

```

ows:reference="xs:string"/>
    <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
        </LiteralData>
    </Input>
    <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
            AutoAdjustMaximumAdjustmentSize</ows:Identifier>
        <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
            AutoAdjustMaximumAdjustmentSize</ows:Title>
        <LiteralData>
            <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:double"/>
                <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
                    </LiteralData>
        </Input>
        <Input minOccurs="1" maxOccurs="1">
            <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
                AutoAdjustMaximumSegmentAngleDelta</ows:Identifier>
            <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
                AutoAdjustMaximumSegmentAngleDelta</ows:Title>
            <LiteralData>
                <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:double"/>
                    <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
                        </LiteralData>
        </Input>
        <Input minOccurs="1" maxOccurs="1">
            <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
                AutoAdjustMinimumIncidenceAngle</ows:Identifier>
            <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
                AutoAdjustMinimumIncidenceAngle</ows:Title>
            <LiteralData>
                <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:double"/>
                    <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
                        </LiteralData>
        </Input>
        <Input minOccurs="1" maxOccurs="1">
            <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
                AutoAdjustSegmentAdjustmentLength</ows:Identifier>
            <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
                AutoAdjustSegmentAdjustmentLength</ows:Title>
            <LiteralData>
                <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:double"/>
                    <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
                        </LiteralData>
        </Input>
        <Input minOccurs="1" maxOccurs="1">
            <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
                AutoAdjustshiftingSegmentswithOneConnectedEnd</ows:Identifier>
            <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
                AutoAdjustshiftingSegmentswithOneConnectedEnd</ows:Title>
            <LiteralData>
                <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
                    <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
                        </LiteralData>
        </Input>
        <Input minOccurs="1" maxOccurs="1">
            <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
                VertexTransferEnabled</ows:Identifier>
            <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
                VertexTransferEnabled</ows:Title>
            <LiteralData>
                <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
                    <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
                        </LiteralData>
        </Input>
        <Input minOccurs="1" maxOccurs="1">
            <ows:Identifier xmlns:ows="http://www.opengis.net/ows/1.1">
                referenceDatasetName</ows:Identifier>
            <ows:Title xmlns:ows="http://www.opengis.net/ows/1.1">
                referenceDatasetName</ows:Title>
            <LiteralData>
                <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
                    <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
                        </LiteralData>
    </Input>

```

```

        <ows:AnyValue xmlns:ows="http://www.opengis.net/ows/1.1"/>
      </LiteralData>
    </Input>
  </DataInputs>
  <ProcessOutputs>
    <Output>
      <ows:Identifier
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLURL1</ows:Identifier>
      <ows:Title
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLURL1</ows:Title>
      <ows:Abstract
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLURL1</ows:Abstract>
      <LiteralOutput>
        <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
      </LiteralOutput>
    </Output>
    <Output>
      <ows:Identifier
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLversion1</ows:Identifier>
      <ows:Title
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLversion1</ows:Title>
      <ows:Abstract
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLversion1</ows:Abstract>
      <LiteralOutput>
        <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
      </LiteralOutput>
    </Output>
    <Output>
      <ows:Identifier
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLschemaname</ows:Identifier>
      <ows:Title
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLschemaname</ows:Title>
      <ows:Abstract
xmlns:ows="http://www.opengis.net/ows/1.1">outputGMLschemaname</ows:Abstract>
      <LiteralOutput>
        <ows:DataType xmlns:ows="http://www.opengis.net/ows/1.1"
ows:reference="xs:string"/>
      </LiteralOutput>
    </Output>
  </ProcessOutputs>
</ProcessDescription>
</wfs:ProcessDescriptions>

```

5.4.2.1 Inputs

The bold identifiers provide the information on the input datasets via WFS GetFeature requests (“**inputGMLURL1**”, “**inputGMLURL2**”). To be able to process the datasets correctly the process requires knowledge about their GML version. This information is provided through two additional inputs (“**inputGMLversion1**”, “**inputGMLversion2**”).

All identifiers decorated with an underscore define the internal behavior of the geometry conflation algorithm – the RoadMatcher library. Section 5.5.1 provides an overview of the input parameters and their meanings.

5.4.2.2 Outputs

The outputs of the geometry conflation WPS are provided as instances of an on-the-fly schema based on GML. It is created within the post-processing (see section 5.4.2.3) and contains the geometry and attributes of the resulting features. The wps:Output with identifier “**outputGMLURL1**” provides a reference to the dataset for download, the identifier “**outputGMLversion1**” provides the reference to the on-the-fly schema for download. To establish the link between the two XML files, the identifier

“**outputGMLschemaname**” provides the schemaLocation as defined in the resulting dataset for the used XML schema.

All elements of the on-the-fly schema are using the same namespace. In particular, this means that both attributes of the source and target dataset are included in the resulting feature if a feature has been identified as a geometric match. The attribute conflation part of the Conflation Architecture is capable of referencing the attributes back to the correct namespace (see section 5.4.1.3 for details).

5.4.2.3 Workflow

A high level workflow of the geometry conflation is illustrated in Figure 4.

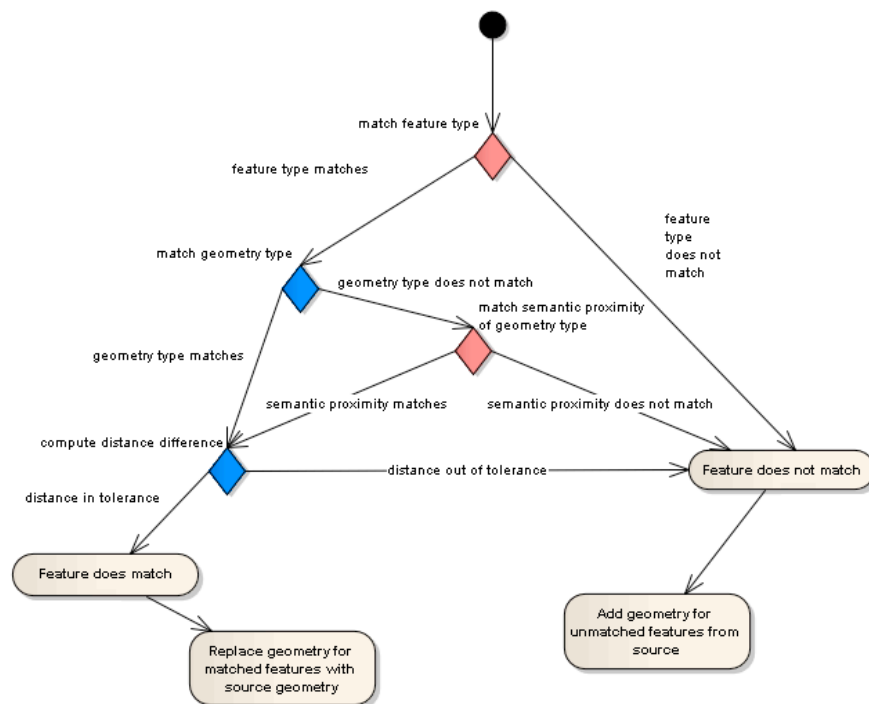


Figure 4 – High-level Geometry Conflation Workflow.

5.5 Conflation Rules

A first task within OWS-9 was the definition of a dedicated set of rules which form the basis for the conflation approach. The outcomes of this task are documented in Annex A of this document. Based on it the processes for geometry and attribute conflation within the OWS-9 conflation have been designed and implemented. The following sections provide an overview on the realization of the defined rules in context of the given use case.

5.5.1 Geometrical Conflation

RoadMatcher, an open source conflation tool, was selected as the base tool to enable the geometrical conflation of linear features. Using RoadMatcher possibly achieves the following objectives in the process of geospatial feature conflation:

- (1) Automatic alignment of linear features: Linear features can be automatically aligned using RoadMatcher by giving tolerances in terms of distance and angle difference.
- (2) Automatic adjustment of linear features: Adjustment of linear features in target coverage is partially automatic if the matched linear features from the reference source inputs are within the given tolerances of geospatial distance (distance and angular difference).

5.5.1.1 Encoding

The encoding of geometric conflation depends on the conflation tool selected. One of the objectives in this phase of OGC Web Service Initiatives is to complete the conflation as much as possible before human manual adjustment by leveraging interoperable components in the Web environment. The selected conflation tool, RoadMatcher, supports automatic batch processing of geometric conflation. The best outcome of the automatic geometric conflation is enabled by properly setting the geometric alignment and adjustment parameters. The encoding of such rules is enabled by properly setting the conflation profile of RoadMatcher. The parameters have to be decided according to the applications and areas. In this phase, the following encoding was found working best with demonstration areas as shown in Listing 6. The distance measure is given in meters. The angular distance is given in degrees. Two coverage inputs are determined at the run time by properly setting manifest as shown in Listing 7.

Listing 6 – Geometrical conflation encoding using RoadMatcher.

```
<?xml version="1.0" encoding="UTF-8"?>
<profile>
  <locked>true</locked>
  <dataset>
    <shortName>f2</shortName>
    <adjustmentConstraint>none</adjustmentConstraint>
    <resultAttributes>area,centerline,condition0,containedI,controllin,
deckLevel,divided,featureCon,flightStri,geointAssu,geoNameCol,geoNameC_1,
length,medianPres,note.memor,oneway,pavementIn,pavement_1,pavement_2,
relativeLe,restrictio,restrict_1,restrict_2,routeDesig,routeMedia,
routeMed_1,routeMed_2,routeTotal,specifiedE,supportedB,surfaceSlo,
surfaceS_1,surfaceS_2,thoroughfa,thorough_1,throughRou,trackOrLan,
uniqueEnti,verticalRe,width,widthOfSec</resultAttributes>
    <vertexTransferFrom>false</vertexTransferFrom>
  </dataset>
  <dataset>
    <shortName>f2</shortName>
    <adjustmentConstraint>none</adjustmentConstraint>
    <resultAttributes>gml_id,permanentI,sourceFeat,sourceData,sourceDa_1,
sourceOrig,dataSecuri,distributi,loadDate,stateRoute,stateCount,
roadClass,isoneway,lowAddress,highAddres,lowAddre_1,highAddr_1,
fullStreet,zipLeft,zipRight,cfccCode,countyRout,onewayDire
  </resultAttributes>

```

```

<vertexTransferFrom>false</vertexTransferFrom>
</dataset>
<autoMatch>
  <minimumLineSegmentLength>2.0</minimumLineSegmentLength>
  <nearnessTolerance>15.0</nearnessTolerance>
  <findStandaloneRoads>
    <enabled>true</enabled>
    <minimumDistance>15.0</minimumDistance>
  </findStandaloneRoads>
  <findMatchedRoads>
    <enabled>true</enabled>
    <maximumDistance>15.0</maximumDistance>
  </findMatchedRoads>
</autoMatch>
<autoAdjust>
  <datasetName>f2</datasetName>
  <maximumAdjustmentSize>100.0</maximumAdjustmentSize>
  <maximumSegmentAngleDelta>20.0</maximumSegmentAngleDelta>
  <minimumIncidenceAngle>20.0</minimumIncidenceAngle>
  <segmentAdjustmentLength>20.0</segmentAdjustmentLength>
  <methodClass>com.vividolutions.jcs.plugin.conflate.roads.
WarpAdjustmentMethod</methodClass>
  <shiftingSegmentswithOneConnectedEnd>true
</shiftingSegmentswithOneConnectedEnd>
</autoAdjust>
<vertexTransfer>
  <enabled>true</enabled>
  <algorithm>com.vividolutions.jcs.conflate.roads.vertextransfer.
ClosestPointVertexTransferOp</algorithm>
</vertexTransfer>
  <precedenceRuleEngine class="com.vividolutions.jcs.conflate.roads.model.
ReferenceDatasetPrecedenceRuleEngine">
  <referenceDatasetName>f1</referenceDatasetName>
</precedenceRuleEngine>
<issueLog />
  <segmentComments editable="true">
  <segmentComment>true</segmentComment>
</segmentComments>
  <onSessionLoad null="true" />
</profile>

```

During this demonstration phase, there are several use cases identified. Corresponding to the use cases of conflation on linear features, two use cases can be done with proper encoding with RoadMatcher.

“Use case 1.1 - (Monterey) (linear features) Conflate the NGA LTDS dataset (target) with the roads, airfields, and rivers (NHD flowlines) from the USGS National Map (TNM) (source) dataset. There are many street level roads in TNM that are not in NGA.”

Encoding rule: The encoding rule for this case is shown in Listing 6.

“Use case 1.1 - (Monterey) (linear features) Conflate the NGA LTDS road geometry & attributes (target) with TNM road geometry & attributes (source) to create an enhanced dataset that has unmatched NGA LTDS matched TNM, and unmatched TNM.”

Encoding rule: The encoding rule for this case is shown in Listing 7.

Listing 7 – Encoding rule for use case 2.

```

<?xml version="1.0" encoding="UTF-8"?>
<profile>
  <locked>true</locked>
  <dataset>
    <shortName>f1</shortName>

```

```

<adjustmentConstraint>none</adjustmentConstraint>
<resultAttributes>area,centerline,condition0,containedI,controllin,
deckLevel,divided,featureCon,flightStri,geointAssu,geoNameC01,geoNameC_1,
length,medianPres,note.memor,oneway,pavementIn,pavement_1,pavement_2,
relativeLe,restrictio,restrict_1,restrict_2,routeDesig,routeMedia,
routeMed_1,routeMed_2,routeTotal,specifiedE,supportedB,surfacesLo,
surfaces_1,surfaces_2,thoroughfa,thorough_1,throughRou,trackOrLan,
uniqueEnti,verticalRe,width,widthOfSec</resultAttributes>
<vertexTransferFrom>>false</vertexTransferFrom>
</dataset>
<dataset>
<shortName>f2</shortName>
<adjustmentConstraint>none</adjustmentConstraint>
<resultAttributes>gml_id,permanentI,sourceFeat,sourceData,sourceDa_1,
sourceOrig,dataSecuri,distributi,loadDate,stateRoute,stateCount,
roadClass,isOneway,lowAddress,highAddress,lowAddre_1,highAddr_1,
fullStreet,zipLeft,zipRight,cfccCode,countyRout,onewayDire
</resultAttributes>
<vertexTransferFrom>>false</vertexTransferFrom>
</dataset>
<autoMatch>
<minimumLineSegmentLength>2.0</minimumLineSegmentLength>
<nearnessTolerance>15.0</nearnessTolerance>
<findStandaloneRoads>
<enabled>>true</enabled>
<minimumDistance>15.0</minimumDistance>
</findStandaloneRoads>
<findMatchedRoads>
<enabled>>true</enabled>
<maximumDistance>15.0</maximumDistance>
</findMatchedRoads>
</autoMatch>
<autoAdjust>
<datasetName>f2</datasetName>
<maximumAdjustmentSize>100.0</maximumAdjustmentSize>
<maximumSegmentAngleDelta>20.0</maximumSegmentAngleDelta>
<minimumIncidenceAngle>20.0</minimumIncidenceAngle>
<segmentAdjustmentLength>20.0</segmentAdjustmentLength>
<methodClass>com.vividolutions.jcs.plugin.conflate.roads.
WarpAdjustmentMethod</methodClass>
<shiftingSegmentsWithOneConnectedEnd>>true
</shiftingSegmentsWithOneConnectedEnd>
</autoAdjust>
<vertexTransfer>
<enabled>>true</enabled>
<algorithm>com.vividolutions.jcs.conflate.roads.vertextransfer.
ClosestPointVertexTransferOp</algorithm>
</vertexTransfer>
<precedenceRuleEngine class="com.vividolutions.jcs.conflate.roads.model.
ReferenceDatasetPrecedenceRuleEngine">
<referenceDatasetName>f2</referenceDatasetName>
</precedenceRuleEngine>
<issueLog />
<segmentComments editable="true">
<segmentComment>true</segmentComment>
</segmentComments>
<onSessionLoad null="true" />
</profile>

```

5.5.2 Attribute Conflation

The process developed within OWS-9 defaults to the “Add-only” use case (“identify attributes from source not present in target and add in target with target schema”, see Annex A).

For unmatched feature in and “add only features” scenario the complete set of attributes is used for newly added features. If the scenario “add only” (features and attributes) is performed, the attributes for unmatched features from source are only added when attribute field is filled in source and unfilled in target.

5.6 Provenance

5.6.1 Conflation provenance

There are two levels of provenance information should be captured in the conflation process. They are dataset level and feature level. At dataset level, provenance captures information about the inputs and outputs of the complete conflation WPS as a whole entity. OGC standards (**OGC 07-045**) and ISO standards (**ISO 19115**) are adopted to support data lineage recording and encoding. For each data source, the following properties will be encoded to provide tractable information on provenance.

5.6.1.1 Identity and Entity Properties

ID (a unique reference to the record in the catalogue)

TopicKeyword (the topic of the content of the resource, some values from a controlled vocabulary or formal classification scheme)

Title (the name of the resource)

Abstract (a summary of the content of the resource)

DataType (the nature or genre of the content of the resource, e.g. dataset, datasetcollection, service, application)

ResponsibleCreator (who produces the resource)

ResponsiblePublisher (who publishes the resource)

LastUpdate (Date on which the record is updated in catalogue)

DataFormat (The physical or digital manifestation of the resource)

DataAnyText (A text description for whole resource in catalogue)

BBOX (a bounding box for identifying a geographic area of interest)

CRS (Geographic Coordinate Reference System)

5.6.1.2 Data Lineage Properties

LineageStatement (general explanation of the data producer's knowledge about the lineage of the resource)

ProcessStepDescription (description of the event, including related parameters or tolerances)

ProcessStepReportName (a name of the report of the processing steps)

ProcessId (an identifier identifying the processing)

SourceDescription (detailed description of the source resource, including information about geographic area and temporal)

SourceProcessedLevel (Processing level of the source resource)

LineageLevel (the times that the resource are processed and recorded with lineage information)

At feature level, what provenance information to be recorded is highly related to the actual dataset and domain under study. The general guideline is the same.

5.6.2 Encoding

ISO 19115 (2003) is adopted as the baseline standard to encode all the provenance information for conflation WPS.

Table 1 – Encoding fields of conflation dataset for Quality Indicator for both dataset and data feature levels.

Name	Status	tag
User Satisfaction Index	extended	short name: userSatInd
DQ_AbsoluteExternalPositionalAccuracy	from ISO	DQAbsExtPosAcc
ChangeInAccuracy	extended	posChanInAcc
PosConflationToolSpecificIndicators	extended	posConToolSpecInd
FeatureDensity	extended	comFeaDen
FeatureDensityImprovement	extended	comFeaDenImp
AttributePopulationDensity	extended	comAttPopDen
AttributePopulationChange	extended	comAttPopChan
ComConflationToolSpecificIndicators	extended	comConToolSpecInd
DQ_ConceptualConsistency	from ISO	DQConcConsis
DQ_DomainConsistency	from ISO	DQDomConsis
DQ_TopologicalConsistency	from ISO	DQTopConsis
DQ_FormatConsistency	from ISO	DQFormConsis
DQ_QuantitativeAttributeAccuracy	from ISO	DQQuanAttAcc
DQ_Non-quantitativeAttributeCorrectness	from ISO	DQNonQuanAttAcc
DQ_ThematicClassificationCorrectness	from ISO	DQThemClassCor

5.6.2.1 Conflation provenance encoding at dataset level

By following ISO 19115 standards, each above element has 5 mandatory and 4 optional attributes, which can be expressed as below in the ISO standards-compatible XML format. These nine attributes are highlighted in red and bold font as shown in Table 2

Table 2 – Example segment of provenance encoding at dataset level

```

<DQ_DataQuality>
  <gmd:lineage/>
  <gmd:report>
    <gmd:DQ_AbsoluteExternalPositionalAccuracy>
      <gmd:result>
        <gmd:DQ_ConformanceResult>
          <gmd:specification>
            <gmd:CI_Citation>
              <gmd:title>Measurements</gmd:title>
            </gmd:CI_Citation>
            <gmd:explanation>
              <gmd:CharacterString>maxium distance</gmd:CharacterString>
            </gmd:explanation>
            <gmd:pass>
              <gmd:Boolean>>true</gmd:Boolean>
            </gmd:pass>
            </gmd:sepcification>
          </gmd:DQ_ConformanceResult>
        </gmd:result>
        <gmd:result>
          <gmd:DQ_QuantitativeResult>
            <gmd:valueType>
              <gco:RecordType>MAXDIST</gco:RecordType>
            </gmd:valueType>
            <gmd:valueUnit>
              <gml:UnitOfMeasure>mm</gml:UnitOfMeasure>
            </gmd:valueUnit>
            <gmd:errorStatistic>
              <gco:CharacterString>error value</gco:CharacterString>
            </gmd:errorStatistic>
            <gmd:value>
              <gco:Record>6.0837689036198705</gco:Record>
            </gmd:value>
          </gmd:DQ_QuantitativeResult>
        </gmd:result>
      </gmd:DQ_AbsoluteExternalPositionalAccuracy >
    </gmd:report>
  </DQ_DataQuality>

```

5.6.2.2 Conflation provenance encoding at feature level

Each feature has an associated metadata attribute. It has a complex inline content with data lineage and quality recordings. Table 3 gives an example for encoding one feature's provenance information.

Table 3 – An example for encoding one feature's provenance information

```

<gmd:MD_Metadata>
  <gmd:dataQualityInfo>
    <gmd:DQ_DataQuality>
      <gmd:scope>
        <gmd:DQ_Scope>
          <gmd:level>feature</gmd:level>
        </gmd:DQ_Scope>
      </gmd:scope>
    <gmd:lineage>

```

```

<gmd:LI_Lineage>
<gmd:source xlink:href="#input"/>
  <gmd:processStep>
    <gmd:LE_ProcessStep id="SampleProcessStep1">
      <gmd:description>
        <gco:CharacterString>SampleProcessStep1</gco:CharacterString>
      </gmd:description>
      <gmd:processInformation>
        <gmd:LE_Processing>
          <gmd:procedureDescription>
<gmd:CharacterString>
ADJUSTED:N;SRCSTATE:Matched(reference);RESSTATE:Inconsistent;
SPLITSTART:N;SPLITEND:N;MANMATCH:N;MANADJ:N;MANADJCNT:0;MATCHORIEN:Y;REVIWED:N;
UPDATETIME:2012-22-15 14:14:19</gmd:CharacterString>
          </gmd:procedureDescription >
        </gmd:LE_Processing>
      </gmd:processInformation>
    </gmd:LE_ProcessStep>
  </gmd:processStep>
</gmd:LI_Lineage>
</gmd:lineage>
<gmd:report>
<gmd:DQ_AbsoluteExternalPositionalAccuracy>
  <gmd:result>
    <gmd:DQ_ConformanceResult>
      <gmd:specification>
        <gmd:CI_Citation>
          <gmd:title>Measurements</gmd:title>
        </gmd:CI_Citation>
        <gmd:explanation>
          <gmd:CharacterString>maximum distance</gmd:CharacterString>
        </gmd:explanation>
        <gmd:pass>
          <gmd:Boolean>>true</gmd:Boolean>
        </gmd:pass>
      </gmd:DQ_ConformanceResult>
    </gmd:result>
  </gmd:DQ_AbsoluteExternalPositionalAccuracy >
</gmd:report>
</gmd:DQ_DataQuality>
</gmd:dataQualityInfo>
</gmd:MD_Metadata>

```

5.6.3 Provenance Realization within OWS-9

The developed WPS process is capable to produce ISO 19115 Metadata at dataset and feature level. Dataset level provenance includes information about the involved process steps (geometry and attribute conflation). The feature level provenance includes all necessary quality indicators to enable the assessment of the resulting dataset by a user expert.

Though, the client had limitations in processing ISO 19115 formatted metadata. Hence, an additional simplified feature level provenance encoding has been defined. Provides and examples of this encoding. The WPS provided two separate processes – one

providing ISO 19115 Metadata and another providing feature level provenance using this simplified encoding. By following this approach, the client software was capable to provide feature visualizations based on the provided provenance information.

Listing 8 – Simplified Provenance Encoding.

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection gml:id="provenance" xmlns:gml="http://www.opengis.net/gml/3.2">
  <gml:featureMember>
    <con:ProvenanceFeature gml:id="usgsToNga10000"
xsi:schemaLocation="http://www.opengis.net/ows-9/cci/conflation
simpleProvenanceEncoding_v4.xsd" xmlns:con="http://www.opengis.net/ows-9/cci/conflation"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
      <con:geometry>
        <gml:LineString gml:id="uuid.e638989e-6e1b-4585-82b4-9affd8909f9d"
srsDimension="2" srsName="urn:ogc:def:crs:EPSG::4326">
          <gml:posList>...</gml:posList>
        </gml:LineString>
      </con:geometry>
      <con:geometrySource>inputTargetDataset</con:geometrySource>
      <con:conflation_startTime>2001-12-17T09:30:47Z</con:conflation_startTime>
      <con:conflation_endTime>2001-12-17T09:30:47Z</con:conflation_endTime>
      <con:processStep_1>Geometry Conflation by GMU WPS</con:processStep_1>
      <con:processStep_1_inputSourceDataset>http://portal.cubewerx.com
</con:processStep_1_inputSourceDataset>
      <con:processStep_1_inputTargetDataset>http://www.interactive-instruments.de
</con:processStep_1_inputTargetDataset>
      <con:processStep_2>Attribute Conflation by 52North WPS</con:processStep_2>
      <con:processStep_2_input>Geometry Conflation by GMU
WPS</con:processStep_2_input>
      <con:inputSourceDatasetFeature>CWFID.TR_ROADS.0.15940
</con:inputSourceDatasetFeature>
      <con:inputTargetDatasetFeature>TransportationCurves1178
</con:inputTargetDatasetFeature>
      <con:nearness uom="m" description="geometry
nearness">0.00000032</con:nearness>
      <con:maxdist uom="m" description="maximum distance">0.000054</con:maxdist>
      <con:trimdist uom="m" description="tirm distance">0.0002</con:trimdist>
      <con:adjsize uom="m" description="adjustment size">0.0003</con:adjsize>
    </con:ProvenanceFeature>
  </gml:featureMember>
</gml:FeatureCollection>
```


6 Lessons learned and open issues

The OWS-9 conflation architecture comprised two separate WPS instances. The geometry conflation WPS was invoked from the overall conflation process WPS. The realization of this configuration showed some drawbacks as the geometry conflation WPS did not provide a well-defined result (e.g. using the target dataset schema and on top of that ISO 19115 Metadata for applied process steps and used rules). In particular, every feature is returned with both source and target dataset attributes as XML elements. For the current use case this approach was feasible, but it has limitations when both datasets share the same attribute names. Assigning the attributes to the correct dataset in an automatic manner would be not feasible. Consequently, this aspect of the architecture provides some options for improvement.

The currently used ProcessDescription does not provide parameters to adjust certain conflation rules. This limits the value of the process as an expert user is not capable of applying tweaks to the dataset conflation.

7 Accomplishments

Within this work an automated approach for dataset conflation has been developed. Conflation rules have been designed which form the basis for the realized WPS processes. The developed processes apply geometry conflation (based on spatial similarity) as well as attribute conflation (based on semantic mediation). A provenance model reflecting the single process steps and the data sources has been developed and realized. Here, a special attention on quality indicators was paid.

8 Recommendations

Currently, the feature level provenance is embedded into the deprecated `gml:metaDataProperty` of the resulting features. Though, the “Revision notes of GML 3.2.1 Implementation Specification” state that a schema that may use metadata on a feature level should define its own elements dedicated to metadata. This additionally allows the schema to require metadata or to make it optional. An example of a dedicated metadata element at feature level is the AIXM 5.1 schema² which defines an `aixm:featureMetadata` element with ISO 19115 Metadata contents. This should be taken into consideration when developing future versions of the TDS and TNM schemas.

The provision of metadata provenance encoded as ISO 19115 Metadata is a very interesting field of work. Here, it can be of great benefit to visually present dependencies of processes, their inputs and individual steps. Figure 5 illustrates an exemplary visualization of such metadata. The data here is not linked to the data used in OWS-9 but

² <http://www.aixm.aero/schema/5.1/>

can provide some ideas on how to visually present provenance and metadata information. More detailed information on provenance visualization can be found in Annex B of this document.

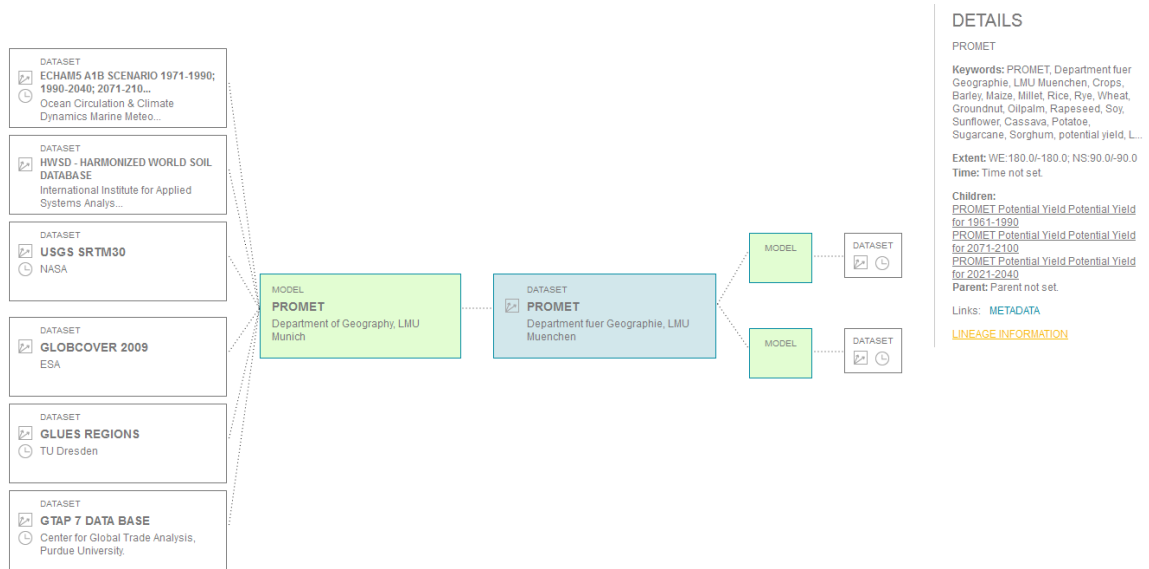


Figure 5 - Exemplary Visualization of Provenance.

Annex A - OWS-9 Conflation Rules

Conflation is the process of unifying multiple separate sources of data into one integrated all-encompassing result, combining the best parts from each source. Conflation is applicable to both raster sources and vector sources and includes geolocation, metadata, and attribution. Digital representations of geospatial features (such as roads, rivers, and forests) vary between databases in regards to alignment, precision, location, completeness, and potentially in geometric representation as well as attribution and topology. The core of the conflation process is identifying and associating the common features across multiple data sources in spite of all these challenges, reconciling the differences between them, and constructing one integrated result. Because of the wide range of integration scenarios, particularly from heterogeneous data sets, rule-based approaches often provide practical solutions. For example, scenarios of data integration in practice could contain: 1) all the unique features and all the unique attributes from the sources being processed, 2) the “best” geospatial representation of features deemed to be common, 3) the combined attribution for features in common, and 4) where values differ for attributes in common, the “best” value likewise must prevail as well.

Disclaimer: This ANNEX is a compilation of generic logical and business rules derived from a number of sources. It does not reflect current or future NGA conflation practices but is intended to support development and testing of various conflation processes and tools as well as demonstrating OGC standards-based data interface and processing concepts to enable future development of validated conflation rules and coordinated standards. The inputs for this paper includes prior conflation studies and development efforts such as the SMARTS business rules validation of 2007-8, the NTA Conflation Report of 2009, the OGC Fusion Engineering Reports, as well as informal discussions with NGA/Source and USGS data stewards and industry conflation tool developers.

Data Preparation and Conflation Pre-processing:

1. Designate source (or subject) and target data (or reference) (e.g. use the National Map (TNM) data (source) to update Topographic Data Store (TDS) data in new TDS dataset (target))
2. Conduct (manual and/or automated) quality assurance (QA) of source and target datasets for internal consistency. Examine and update topology for consistency and correct logical errors and gaps (overshoots, undershoots, absent metadata, etc)
3. Transform schema to comparable elements (elevation and accuracy fields (ft or m), coordinate reference system, common data structures
 - a. For TNM and TDS, build on semantic mediation for OWS-8

- b. For datasets not semantically mediated or aligned, create and apply common mapping table techniques (e.g. for OSM, build new semantic mediation or HALE transformation)
- c. Use common schema and metadata standards (for TDS as target, use NMF and NFDD)

Define Conflation Approach:

1. Add only (features only)
 - a. When used: target is mature and largely complete and only want to add features not presently in dataset
 - b. Compare features in source to features target, identify features from source not present in target and add features (as attributed in source) to target database with target schema
2. Add only (features and attributes)
 - a. When used: target is somewhat mature and largely complete but lacks features and detailed attribution
 - b. Complete comparison and addition from add only (features only) process
 - c. Compare features in source to features target for matched features , identify attributes from source not present in target and add in target database with target schema
3. Update (replace features and attributes)
 - a. When used: target is fairly comprehensive but somewhat dated and source is less populated but more current and accurate
 - b. Compare features in source to features target, identify features from source not present in target and add features (as attributed in source) to target database with target schema
 - c. Compare features in source to features target for matched features , add features and attributes not present in target and overwrite matched features to include attributes
4. Qualitative update (select best feature and attributes)
 - a. When used: datasets are similar in quality and density with both gaps and overlaps in features and attributes
 - b. Compare features in source to features target, identify features from source not present in target and add features (as attributed in source) to target database with target schema
 - c. Compare features in source to features target for matched features , apply conflation rules to determine best quality features and populate target
5. Vector to Raster conflation

- a. When used: Vector dataset is richly attributed but geometry is dated and/or inaccurate.
- b. Compare vectors in dataset to raster data or imagery.
- c. Update and align geometry of existing features for improved accuracy while maintaining attribution.
- d. Identify and add/attribute features in raster/image source not present in the dataset.

Conflation Rules for determining feature matches, selecting geometry, selecting attributes, and capturing provenance

1. Conflation techniques vary greatly and could include visual data inspection, automated rule discovery algorithms, comprehensive rule development and definition, and/or iterative rule refinement. The degree of automation and rigorous rule development should consider the data quality of datasets and the conflation approach selected.
2. Feature matching:
 - a. Features in target and source datasets can be considered possible conflation candidates based on the degree of similarity and confidence associated to those feature geometry in each dataset. The user or tool should invoke any or all of the following rules to declare matches for conflation of geometry and attributes. As a minimum, feature type rules and distance rules should be active. Some rules may also allow parameter setting to expand or limit potential matches.
 - b. Feature type
 - i. Feature type must be exact match (e.g. both roads) AND/OR
 - ii. Feature subtype must be exact match (e.g. both primary roads) AND/OR
 - iii. Feature type or subtype must be of similar class (e.g. road matches trail; primary road matches secondary road)
 - c. Distance difference
 - i. Match based on how close features are spatially(e.g. the distance between two features are within the thresholds for a match)
 - ii. Examples of matching process include RMSE (points), Hausdorf distance (polylines), epsilon band (polylines), area overlap (polygons),
 - iii. Specify if exact match is required or if match within distance tolerance is acceptable.
 - iv. Set parameter for acceptable feature match (design as user-defined input threshold (e.g. within 10 or 25 meters)
 - d. Height difference (is the height data present and within the specified ranges)

- i. Match based on how close height of features (are two buildings or two vertical obstructions same or different)
 - ii. Set parameter for acceptable feature match (design as user-defined input threshold (e.g. within 10 or 25 meters))
 - e. Feature Geometry type
 - i. Geometry type must be the same (points to point; line to line; area to area only) OR
 - ii. Geometry type may be different IF semantic proximity matches (degree to which qualitative descriptions and attributes converge or diverge: e.g. point and area representation of same function (e.g. water treatment facility; feature geometry is different based on level of generalization and scale). This relies heavily on data-specific mappings or manual review. (Consider if semantic mediation can enhance as a viable approach).
 - f. Determine matched and unmatched features (features in both, features in source only, features in target only)
- 3. Geometry selection (alignment and adjustment)
 - a. Select based on conflation approach (add, update, or quality update)
 - b. For add only (features only), add geometry for unmatched features (with attributes) from source and disregard matched features
 - c. For add only (features and attributes), add geometry for unmatched features from source
 - d. For update, add/overwrite geometry for unmatched features (with attributes) and matched features from source
 - e. For qualitative update, apply geometry rules to select from provided matched geometry. Rules can be considered possible conflation based on the quality of feature and metadata including source type, source date, accuracy, and other inputs. As was the case for feature matching, some or all of the rules may be applied. Rule ordering and weighting should also be established to capture the relative importance of accuracy, currency, quality, and other criteria.
 - i. The geometry with the newer is better for Source Date/Time (this may direct comparison if both datasets indicate image source date and time but more complicated if source date/time entry refers to when analyst populated dataset and not to when source info collected)
 - ii. The geometry with the smaller error in vertical accuracy is better
 - iii. The geometry with the smaller error in horizontal accuracy is better

- iv. The geometry with greater resolution or detail is better than one more generalized
 - v. The geometry type with more detail is better (e.g. an Area is better than a Line or Point and a Line is better than Point)
 - vi. The geometry with qualitatively better source type is preferred (e.g. features extracted from high resolution imagery are better than features from low resolution imagery which are better than features produced by trusted partners which are better than features purchased from industry which are better than features volunteered by unknown sources) [Comment: this qualitative hierarchy may vary for specific cases and data (e.g. OSM often provides improved geometry). Validation is required. We should also consider allowing this to be a user-defined rule.]
4. Attribute selection (alignment and adjustment)
- a. Select based on conflation approach (add, update, or quality update)
 - b. For add only (features only), add geometry for unmatched features (with attributes) from source and disregard matched features
 - c. For add only (features and attributes), add attributes for unmatched features from source and then add attributes for matched features only when attribute field is filled in source and unfilled in target
 - d. For update, add/overwrite attributes for unmatched features (with attributes) and matched features from source
 - e. For qualitative update, apply attribute rules to select from available attributes in source and target datasets. Rules can be considered possible conflation based on the quality of feature and metadata including source type, source date, accuracy, and other inputs. As was the case for feature matching, some or all of the rules may be applied. Rule ordering and weighting should also be established to capture the relative importance of accuracy, currency, quality, and other criteria.
 - i. Populated attributes are better than null values.
 - ii. The attribute with the newer is better for Source Date
 - iii. The attribute with greater detail or precision is better than one more generalized
 - 1. Specific road type (primary, secondary, trail, etc.) is better than generic classification (road)
 - 2. Attributes with calculated or derived values may be assessed based on relative comparison of values (e.g. smaller Absolute Circular Error (ACE) is preferred) or by comparison of

calculation technique (assess value of different means of deriving Absolute Horizontal Accuracy (AHA)).

- iv. Security and releasability fields must default to the more stringent requirement. Security should be retained for the level (layer, feature, attribute) from which it was maintained. (Coordination with OWS-9 SSI thread required)
- v. The attribute with qualitatively better source type is preferred. A subjective assessment may be included to account for user confidence in inputs derived for differing sources (e.g. features extracted and attributed from high resolution imagery, features from low resolution imagery, features produced by trusted partners, features purchased from industry, and features volunteered by unknown sources)
- vi. When relative attribute quality is Undecided, the attribute from the feature with the best Geometry or horizontal accuracy is preferred.
- vii. In some cases, the output may consider retaining both source and target attributes (e.g. road name could be mapped to RoadName1 and RoadName2 in cases where both names may be in common use or both are desired for historic lineage).

5. Provenance

- a. Provenance should include the lineage and pedigree of the data, the process of conflation, the quality and the accuracy and uncertainty of the input data and sources as well as the accuracy and uncertainty of the output dataset.
- b. The decision criteria and process should be documented in metadata or by reference or context.
- c. Design consideration should include documenting provenance at the layer level or feature level or attribute level or in some combination. Typically, accuracy in a dataset, if available, is generalized for an entire data product class, and represents the minimum predicted accuracy (or maximum predicted error). Selected features may have better local accuracy than the stated accuracy at the layer level. Where possible, provenance of sensor models and extraction methods used should be retained at the feature level to make available more rigorous methods to determine local accuracy for raster and vector data. When conflating attributes, retain provenance of selected value.

Post Conflation Processing

Conduct automated or manual quality assurance (QA) of conflated target datasets for internal consistency.

- a. Examine and update topology for consistency and correct logical errors and gaps (overshoots, undershoots, joins, etc)
- b. Verify conflated set preserves topological relationships (e.g., a building stays on the correct side of a road).
- c. Examine metadata for absent and unpopulated fields and apply rule-based autofill processes.
- d. Resolve duplicative or undecided matches

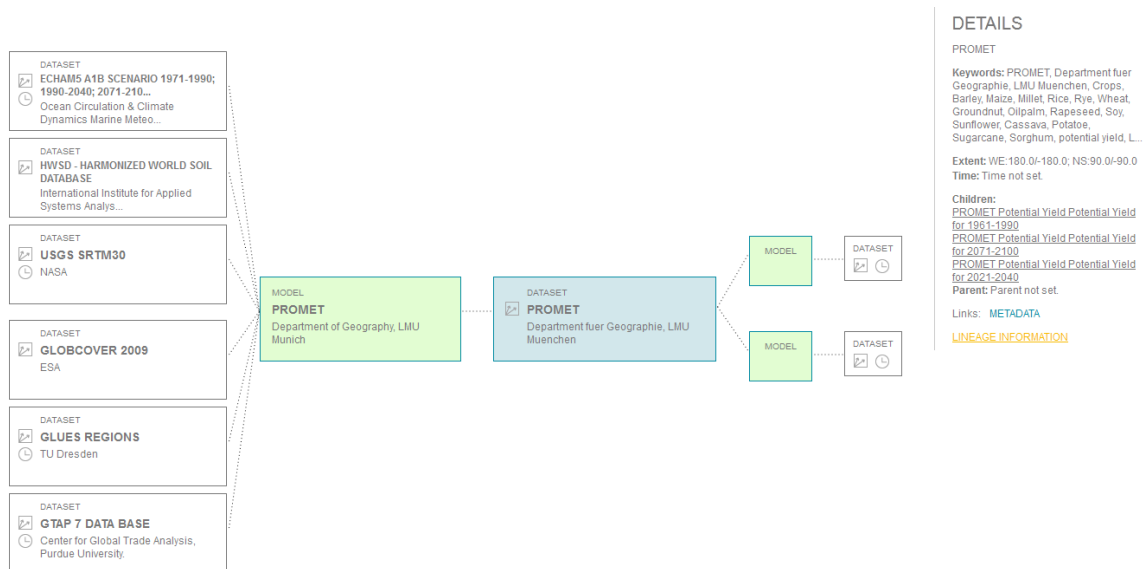
Annex B – Provenance Visualization

Provenance metadata informs the user about the history of a data product, including processes, transformations and maintenance during life of a data set and the source data used to create the dataset. The term provenance is understood here synonymously with the terms lineage or pedigree. Lineage is established as a term in the geoinformation domain as it is defined in the ISO/TC211 (2002): 19113 - geographic information - quality principles:

“Lineage shall describe the history of a dataset and, ..., recount the life cycle of a dataset from collection and acquisition through compilation and derivation to its current form.”

Provenance can be stored detailed at the feature or even attribute level or at the level of complete datasets, which is considered here. The metadata element for lineage as defined in ISO19115/ NMF Part 2 provides the two main elements ProcessStep and Source. Each ProcessStep provides elements for a description, process documentation, software reference including an identifier, run time parameters, rationale, time of processing and processor. Source provides the elements description, scale denominator, reference system, citation reference, extent, processing level and resolution. Further the elements provide associations to link processes to the corresponding source data and vice versa.

This rather simple model is sufficient to provide an overview about the processing a dataset originates from; nevertheless a detailed description of workflows is not possible. This is particularly because the order of multiple successive processes is not explicitly represented. For an explicit representation each processing step must result in an interim dataset, which is not always meaningful and would result in extensive amounts of metadata.



Tabular or listed representations of provenance metadata, as they are commonly implemented in geodata catalogue user interfaces, do not support user-friendly navigation or browsing through the metadata. Also the interactive analysis and comparison of metadata sets is usually not possible. To improve this and to better demonstrate provenance metadata contents a lightweight web client for visualizing the lineage information as tree has been implemented.

<http://geoportal.glues.geo.tu-dresden.de/geoportal/Applications/metaviz.html>

The web client is based on Java and JavaScript. The tool supports users finding out if data fits for his or her purposes by illustrating the data's origin in an intuitive way. With the tool the provenance of a dataset is going to be represented in a tree-like graph that connects the dataset (blue box in the above screenshot) to the process it originates from (green box) as well as the source datasets of these processes (white boxes). The focus of the visualization can be either on an input dataset, a process or an output dataset, such that the origin or the use of a dataset can be analyzed. Relevant metadata of the data set, such as parent-child-relationships to other data sets and data series, publication references and in particular quality information are listed next to the visualization. If the metadata contains distribution information of visualization or download services these links can also be accessed from the provided user interface or will be used to access the data in further applications, such as visualization clients. Further the tool allows the user to navigate through the provenance tree by selecting a source data set or a process/workflow step. Beside provenance information it also allows to visualize the usage of a data set; i.e. if a data set is referenced as source in the provenance information of other data sets in the catalogue. Clicking the triangle at the right top switches between lineage and usage view of a dataset.

The client is requesting ISO 19115 conformant lineage data from OGC compliant CSW 2.0.2. Different catalogs can be registered with the client. The client can be called in a parameterized way by defining the data set the provenance information should be visualized for, or not parameterized which results in a table view of all data sets with provenance information available in the registered catalog.

Links and Examples:

<http://geoportal.glues.geo.tu-dresden.de/geoportal/Applications/metaviz.html>

PROMET data of Chair of Geography and Remote Sensing, LMU Munich:

<http://geoportal.glues.geo.tu-dresden.de:8080/MetaViz/detail.jsp?id=glues:lmu:metadata:dataset:promet>