RESEARCH ARTICLE

Open-data-driven embeddable quality management services for map-based web applications

Alexey Noskov and Alexander Zipf

Institute of Geography, Heidelberg University, Heidelberg, Germany

ABSTRACT

Various map-centered web services facilitate citizens’ lives. Web-map applications exist for many years already. Due to simplification and improvement of technologies supporting WebGIS, map-based services become more popular and important nowadays. Data quality assurance for such services is a significant challenge. Since many of such applications intensively use open data, approaches focused on open solutions are required. This work proposes a data-quality concept, which is based on intrinsic and comparable approaches. OpenStreetMap (OSM) allows intrinsic data evaluation. Moreover, it is used as a reference dataset for quality assessment of public-sector-information Open Data layers. Equidistant point (EDP)-based statistics enables to filter out low-quality Open Data features. A data-type model carries out the inventory of OSM data. The comparison of raster web-map tile file sizes and calculation of a simplified data quality indicator make it possible to specify acceptable data quality levels. Embeddable instances of quality assurance web services incorporate data features with acceptable quality. This work provides all required software and data for the deployment of such services under liberal licenses. Concrete instructions allow users to adopt the proposed solutions for their platforms. Some generic use cases illustrate the advantages of the introduced shared web services.

ARTICLE HISTORY

Received 31 December 2018
Accepted 28 February 2019

KEYWORDS

Data quality; big spatial data; WebGIS; REST API; computer vision

1. Introduction

Nowadays, various map-based applications are available online. Alenezi, Tarhini, and Masadeh (2015) and Bertot, Gorham, Jaeger, Sarin, and Choi (2014) have mentioned that the data quality is one of the most important factors of public sector web services success. In this work, we introduce a data quality management system and describe open-data-based quality assurance processes for map-centered web-applications solutions.

Rousell, Noskov, and Zipf (2018) have presented a data quality concept developed for WebGIS solutions. The work is conducted in the frame of the WeGovNow (We Government Now) project funded by EU Horizon-2020 program. WeGovNow (WGN) (Boella et al., 2018) is a research and innovation action focused on civic participation in local government aiming at using state-of-the-art digital technologies in community
engagement platforms to involve citizens in decision-making processes within their local neighborhood. The concept is based on intrinsic and comparable data quality assessment processes. Open datasets (mainly, OpenStreetMap (OSM)) are utilized in the frame of the work.

The mentioned solutions do not require authoritative ground-truth reference datasets, and only open data are used. Moreover, a simplified data quality Indicator has been proposed as a core data quality indicator. We extend this indicator in the present article by the information extracted from the OSM raster web tiles.

Intrinsic solutions have been applied to calculate OSM data quality levels for every tile area. For this, we aggregate various parameters delivered as a part of OSM datasets. In this stage, the proposed evaluation utilizes only intrinsic properties without the involvement of external datasets. Then, external datasets facilitate the definition of acceptable OSM data quality level. In other words, comparable data quality assessment processes are involved in the definition of conformance OSM data quality levels. For this, we utilize a raster tile assessment approach.

Noskov (2018a) has proposed a practical approach for extracting the information from raster web tiles and comparison of data provided by various web tile services (e.g. OSM and Google Maps (GM)). The mentioned work considers two aspects: sizes of PNG files and points extracted by computer vision processes. In the present work, we use sizes for comparison of OSM and Google Maps (GM) tiles. An edge detection algorithm facilitates the extraction of information from OSM raster tiles. The main advantage of the size-based solution is that the size of a tile can be obtained from headers without downloading a PNG file. It dramatically speeds up processing and allows specifying the minimally acceptable quality of OSM data in a pilot site.

Afterward, we utilize evaluated OSM data for the comparative assessment of other datasets. For this, we exploit the fact that OSM data comprise similar data feature type provided by other datasets. Various rules can be applied for quality assurance. For instance, if a data layer provides higher information quality in comparison with corresponding data extracted from OSM and OSM data provide an acceptable quality of data in a considering area, it indicates the high quality of an examined data layer.

We combine the mentioned solutions to prepare a spatial database. For this, we utilize the Integrated Geographic Information System Tool Kit (IGIS.TK). IGIS.TK has been introduced by Noskov (2018b). In the present work, we use some data conversion, and quality assessment scripts resided in “c” (conversion) and “q” (quality) directories of the toolkit correspondingly. IGIS.TK utilizes a shared library assembled into the Tiles Common Framework (Tiles.CF) project. The framework is used by the Geo-Spatial Data Repository (GSDR) project (Zipf & Noskov, 2018). GSDR is a WebGIS consisting of three main components: spatial database, a universal RestAPI (Masse, 2011) instance, and an advanced HTML5 frontend.

In the present work, we use two first components for the implementation of an embeddable system for the interactive improvement of data quality. In this paper, we describe concepts and processes applied to the preparation of a spatial database. A database is a core component of the proposed embeddable systems. A database comprises data retrieved from OSM, Open Data and raster web tiles (OSM and GM). A light-weight RestAPI uses a database for “on-the-fly” improvement of user’s input. In the conducted earlier research, only one pilot site was covered, the
current article considers four pilot sites: Southwark (London, UK), San Dona di Piave (Italy), Turin (Italy) and Heidelberg (Germany). The Heidelberg pilot site is not fully covered; only OSM and raster tile data have been processed for its area. Heidelberg is not a WGN’s pilot site; it is involved as an additional pilot site, because of authors’ deep knowledge respecting the area (especially, in the OSM context). We have archived the generated datasets (WeGovNowConsortium & OpenStreetMap-Contributors, 2019) and collected specific software (Noskov, 2019a) required for the deployment of the proposed embeddable solutions; further, in the text, we will address to them as “archiving data” and “archiving software,” correspondingly. The archiving software consists of the three following projects: Noskov (2019b, 2019c, 2019d).

The rest of the article is organized as follows. Section 3 describes source data. Section 4 considers the methods and software for data processing. The results are discussed in Section 5. Moreover, it discusses a data model, provides the inventory of resulting datasets and demonstrates the concrete steps required for the deployment of the introduced solutions. Section 6 concludes the paper.

2. Related work

In this section, we describe recent achievements in quality of service (QoS) assurance and data quality related to web services and open data. QoS solutions are worth to be considered because many solutions offer aggregated quality metrics. Similar approaches are developed in the present work. Many such works attempt to run the web services. Al-Masri and Mahmoud (2007) proposed a Web Service Relevancy Function (WsRF) to measure the relevancy ranking of a particular web service based on the users’ preferences and some metrics like response time, throughput, availability and cost. A mathematical matrix is utilized. The method is suitable for QoS metrics that have real numbers. However as security is often described by fuzzy terms, the application of this method is limited.

Chan, Chieu, and Kwok (2008) use a singular value decomposition technique and a user weighting system to find web services correlations. This facilitates the selection of web services. Moreover, web services ranking under multi-criteria matching conducts accurate web service selection and assigns a dominance score to each advertised web service. Additionally, other similar solutions define a business-focused ontology to enable semantic matchmaking in open cloud markets.

Skoutas, Sacharidis, Simitsis, Kantere, and Sellis (2009) proposed the concept of QoS. It considers the security as part of QoS requirements. According to the authors security requirements such as the strength of a cryptographic algorithm, the length of a cryptographic keys, security functions, confidence of policy-enforcement and the robustness of an authentication mechanism can be specified and measured as the quality of security services. The web services are measured by a satisfaction function, which covers both measurable and non-measurable characteristics. For instance, the property of confidentiality is measured by combining the encryption algorithm, key length and key protection used. The web service that has the greatest value in satisfaction function will be chosen.
According to Sadiq (2013), data quality has been investigated by researchers and practitioners. Data quality dimensions (e.g. accuracy, completeness, consistency) play a key role in the definition and measurement of data quality. Data quality management projects are focused on the data quality evaluation (Batini, Cappiello, Francalanci, & Maurino, 2009). Data quality is assessed against either product specification or user requirement (Loshin, 2001).

Rousell et al. (2018) consider the data quality problem in the scope of modern WebGIS application providing a general concept based on the combination of intrinsic and comparable quality assessment. In this work, we follow the proposed solutions.

3. Source data

As mentioned earlier, four pilot sites are covered. Figure 1 shows them. Areas of pilot sites in square kilometers are as follows: San Dona di Piave – 111, Turin – 160, Southwark – 167, and Heidelberg – 391. The concrete boundaries of the pilot sites are provided by *.poly files of the archiving datasets (WeGovNowConsortium & OpenStreetMapContributors, 2019). Various OSM tools use poly files for clipping data from the OSM data dump. We utilize OSM-history-splitter (Körner, 2017) for this. In this section, three data sources will be considered: OSM, OpenData and OnToMap (Ardissono et al., 2017). All data are collected into a single data file. This file is a core GSDR’s spatial database. For every pilot site, we prepare a minimized light-weight database. These databases are designed for the FirstLife (Antonini et al., 2016) servers. FirstLife and OnToMap are components of WGN; they provide several shared spatial components (source of public-sector-information data layers, users’ contribution logging, map tiles web server and common map widgets).

Figure 1. Pilot sites.
FirstLife and OnToMap are delivered as separated instances for every pilot site; for every instance, we provide light-weight database files and deploy an embeddable service for users’ input improvement.

3.1. OpenStreetMap

The archiving dataset contains prepared clipped data (see files with the extension *.bz2 in Table 3). The latest XML OSM planet dump (OpenStreetMapContributors, 2019) has been downloaded. We use the Spatialite (Furieri, 2019) database management system. OSM XML data are converted to the SpatiaLite format. For this, we, first, convert an XML data file to a GeoJSON file by the osm2geojson tool. Second, the c/geojson2spalite.tcl script (Noskov, 2019c) converts GeoJSON data to SpatiaLite. Currently, the script does not work with big geojson files. Thus, the processing of large data files requires splitting such files. We use the geojsplit (Wood, 2016) for this (maximal length of output chunks is 2000 geometries). c/geojson2spalite.tcl can also digest ESRI shapefiles. Since it uses the OGR (Warmerdam & Contributors, 2019) programming library, the script can cover various popular vector data formats. In the output database, we use OSM-like key-value data model; thus, all OSM items are preserved.

3.2. Open data and OnToMap

Open Data are data layers provided for the WGN project by municipalities or obtained for public websites under Open Data license. The archiving data comprise original data layers (see the opendata_sandona.zip, opendata_southwark.rar and opendata_turin.rar files). One can notice that these data are ESRI shapefiles. Because they use an attribute table to store attribute information, we converted attribute data into a database using a key-value format where the key is a column name. Moreover, Open Data layers are delivered in various coordinate systems; thus, all layers are transformed into EPSG-4326 coordinate system (Warmerdam, 2006).

OnToMap is a shared service of WGN. It collects and provides shared datasets including logging users activity for all WGN components and services. Open Data layers have been refined and imported into OnToMap’s database. OnToMap provides data in the JSON and GeoJSON format through the RestAPI interface. The complete database contains both Open Data and data derived from OnToMap; light-weight databases comprise refined OnToMap data only.

4. Methods and software

In this section, we introduce an “ad-hoc” data quality assurance methodology required for the preparation of embeddable databases. As introduced in Rousell et al. (2018), we evaluate data intrinsically and comparably. The mentioned work provides the motivation and advantages of such an approach. The main point is that ground-truth reference datasets are not available in the frame of our project. Thus, we verify data intrinsically using internal properties of the available spatial information and comparably exploring similar objects provided by different data sources. Furthermore, in the present work, we compare the information provided by two sources of raster tiles.
In the present work, we follow the KISS principle ("keep it small and simple"). According to it, we use only simple and straightforward solutions from the previous works. For instance, only the line statistics are derived from the Data-Type model (Noskov et al., 2019), because tag statistics requires quite complex and labor analysis. The line statistics can be easily interpreted and normalized for the comparison of heterogeneous pilot sites.

Moreover, from the raster tile analysis (Noskov, 2018a), we mainly focus on the size-based approach. Edge detection algorithms are applied only for the OSM tiles evaluation. In order to follow the KISS principle, we formulate several simple rules-of-thumb. Such rules are based on a conducted intensive visual and statistical analysis. One can characterize such analysis as “ad-hoc,” which means that it has been conducted for specific pilot sites, data and requirements. Despite this, we believe that the rules can be prospectively applied for other conditions with a minor refinement.

For the formulation the rules-of-thumb, several stages are carried out. First, we calculate equidistant point (EDP, see Figure 2) statistics for brief preliminary evaluation of the OnToMap data. This allows filtering out low-quality data layers. Second, we order pilot sites according to the quality of OSM data using Data-Type model (the line statistics). We use it for further specification of conformance quality levels. Third, we calculate the simplified data quality indicator, which facilitates the preparation of data for the resulting databases.

For the implementation of the proposed methods, we have developed a number of scripts delivered as either the archiving software assembly and dependencies (Noskov, 2019a) or three independent projects (Noskov, 2019b, 2019c, 2019d). Table 1 provides the information regarding the archived files. The content of the first file is described in A. The second file comprises scripts required for the data conversion and various quality

Figure 2. A map of one-meter equidistant points (red points) of OpenStreetMap features. Black lines are OSM polygons’ boundaries; green lines are OSM polylines.
indicators calculation. `tilescf.tar.gz` is a collection of common libraries utilized by Tiles.CF and GSDR. The fifth file comprises a shared library and a Tcl (Flynt, 2012) extension of the GNU make (Stallman, McGrath, & Smith, 2002) framework. Other files in the table are extracted from `nkov.tar.gz`. They have required for the deployment of the proposed solutions. In A, we describe how to use the files for the deployment of the prepared software.

### 4.1. Equidistant point statistics

Doytsher and Noskov (2015) have proposed EDPs statistics for the integration of spatial datasets. EDP-based approaches are popular in computer vision research (e.g. in Belongie, Malik, & Puzicha, 2002; Ma & Latecki, 2011), the detection and comparison of shapes in raster (photo) images. We have found that such solutions are very useful for the comparison of heterogeneous spatial data layers. They can be applied for all geometric types and intuitively clear and understandable. For the preparation of EDP datasets, we use the `tilescf/segment.tcl` library of the `nkov.tar.gz` archive.

Here, we use EDP for detecting low-quality OnToMap data layers. The central principle is as follows. EDPs are calculated and saved to a separate data layer. Point data are taken unchanged. EDPs reside online features. Polygon objects are treated as line features; for this, boundaries are processed. Figure 4 illustrates EDP data for the National Theater (nearby Waterloo Bridge) area. We calculated EDP for two data types: OSM and OnToMap. For OSM, a 1-m interval between points is applied. In order to reduce a time for further processing, a 3-m interval is used for the OnToMap data.

Now, we can calculate the shortest distances between points of two datasets. It allows us to define an average shortest distance from an OnToMap feature to OSM objects. This approach enables to mark outstanding (in a negative sense) data layers. Notice that all points, nodes and vertices are included into EDP datasets. Moreover, EDP facilitates evaluation of the correctness on the import of Open Data into OnToMap. For this, we prepare 1-m EDP for both datasets and detect Open Data EDP without correspondences in OnToMap.

As mentioned, for comparative evaluation of the OnToMap data, we define the shortest distance from corresponding points to the closest OSM feature (i.e. point, line or boundary). Then, for each layer, we aggregate a set of parameters: number of EDP, minimal, maximal, average, median values and standard deviation of shortest distances.

**Table 1. Archiving software: an embeddable instance of GSDR, IGIS.TK, Tiles.CF and scripts for the software deployment.**

| #  | File name       | Description                                           | File size |
|----|-----------------|-------------------------------------------------------|-----------|
| 1  | emgsdr.tar.gz   | Source code of the embeddable system                  | 10 Kb     |
| 2  | idistk.tar.gz   | Source code of IGIS. TK                               | 3.8 Mb    |
| 3  | libmake.tcl     | A library extending GNU make                          | 707 B     |
| 4  | Makefile        | GNU make main file                                    | 3 Kb      |
| 5  | nkov.tar.gz     | A general-purpose dependency library                  | 4 Kb      |
| 6  | tclshc          | A tuned version of the Tcl shell                       | 119 B     |
| 7  | tilescf.tar.gz  | A library for geo-tiles and EDP processing            | 10 Kb     |
The number of EDP represents the quantity of geometric information provided by a layer. Minimal values allow identifying matched features and distinguishing outstanding objects; in most cases, it is about zero. Maximal values enable to define an object with outstanding parts. Average and median values provide a common impression regarding matching with OSM data. Standard deviation values show homogeneity/heterogeneity of EDP statistics regarding a data layer. All these parameters are aggregated for a data layer. In the resulting data various of the acronyms described in Table 2. A data layer is a set of spatial features of a certain type of pilot site. We name data layers according to the convention: DataType (it could be either “OSM” or “PSI”), DataSource (“OTM”, “ODT”, or “CUR”), PilotSite (“SD”, “TR”, “SW”, “HD”) and DataClass. Data class is a set of spatial features grouped thematically (e.g. school, cinema, hospital). In OSM terminology, PSI more or less corresponds to the “amenity” term; the correspondence is not exact. The “PSI.OTM.SD.School” layer name means Public Sector Information (PSI) derived through the OnToMap’s API for the San Dona di Piave pilot site comprising schools objects.

### Table 2. Acronyms used for in the resulting datasets.

| Acronym | Meaning |
|---------|---------|
| OSM     | OpenStreetMap |
| PSI     | Public sector information |
| ODT     | Open data |
| CUR     | Current OSM dataset |
| SD      | San Dona di Piave |
| SW      | Southwark |
| HD      | Heidelberg |
| TR      | Turin |

4.2. Data-type model

As mentioned, we utilize line statistics from Noskov et al. (2019). Line statistics are provided for lines of OSM-XML data. The number of lines and the number of characters are the two main parameters. The overall number of string lines (newline separated) of a pilot site OSM data (see *.bz2 files of the archiving dataset) is used for the normalization of other parameters. We define the number string classes for evaluation of OSM original data. In contrast to the mentioned work, we use the latest OSM XML data, not full-history data.

Starting (sblank) and ending (fblank-) are first two classes. The former reflects the complexity of an XML objects tree. Starting spaces (spaces or table) are used for indents. Ending spaces usually mark imperfection of data writing processes. Thus, the former indicates richer datasets. Hence, it is a “positive” indicator (i.e. the bigger value represents higher data quality). The latter is a “negative” indicator (i.e. the bigger value represents lower data quality). For such classes, we use a minus (“-”) suffix.

The XML format adopts the following special characters: less, more, slash, equal and space. Less and less-slash open an XML instruction. More character closes an instruction. The equal sign joins key-value attributes; spaces separate words. First, the XML starting instruction characters frequency is calculated: less followed by a word (stags), less-slash (ctags) and less not followed by a word (less-), less detects errors. The ending instruction
are covered by the following classes: slashmore and more. Spaces are split into two classes: one space character followed by non-space character (mblanks1) and several neighbor space characters (mblanksmore-). Moreover, the frequency of equal signs is reported.

Five classes outside XML attribute scope (encapsulated in double quotes) are distinguished: ASCII digits (noatrs09), ASCII lower-case letters (noatrsaz), ASCII upper-case letters (noatrsAZ), the rest ASCII characters (noatrsASCII-) and other characters (noatrsany-).

Finally, atrs classes (i.e. string classes inside the attribute scope) are reported. atrs-blanks reports spaces. Three ASCII classes are as follows: digits (atrs09), lower (atrsaz) and upper-case (atrsAZ) letters. Non-ASCII symbols should not appear in OSM XML data files. Thus, all such classes have a “negative” context. Non-ASCII letters in lower (atrslow-, e.g. Cyrillic letters -) and upper-case (atrsup-, e.g. Cyrillic letters -) are reported. Then, other letters without the upper or lower case are reported (atrsalpha-). Furthermore, non-ASCII digits are detected (atrsdigit-). Finally, punctuation (atrspunct) and printable (atrsgraph) characters are defined. All rest characters belong to the class atrsany-. It discloses imperfections of an OSM XML data types.

For each class, we order pilot sites according to the normalized by a lines number of characters. A pilot site obtains scores ”1,” ”2,” ”3” and ”4” from lower to higher values. For classes with the suffix “-,” the pilot sites obtain scores is the descending order. Total summarized scores allow ordering pilot sites. The bigger score indicates a higher quality of data set. Since the OSM project uses an advanced tools for XML data formatting according to the same rules and data model, data files should not contain unnecessary useless spaces and other elements. Thus allows us to use trivial parameters, like number of lines and number of characters, for the quality assessment. Further combining multiple parameters enables to calculate aggregated quality values.

The Data-Type model is applied only for OSM data. The IGIStk/c/osh2sql.tcl -i map.osm -l linestats.txt script of the archiving software calculates the line statistics. Users can run the IGIStk/q/inventosmd3.tcl -l inestats.txt -o/tmp/output.HTML command to present the results as an HTML document comprising a table and a chart.

### 4.3. Raster tiles processing

Noskov (2018a) have proposed solutions for automatic analysis of raster tiles data. OSM and GM data have been considered. Two types of approaches have been introduced. The first is based on the processing of raster tile sizes. This method is quite simple and does not require actual retrieval of raster tiles, but it is not accurate and provides a general overview without details. The second is based on the Canny edge algorithm for the detection of linear elements on raster images. This method is more accurate, but it requires the actual retrieval of raster files. Since the most map tiles providers restrict the massive retrieval of such data, it could be quite problematic to implement the method for large areas. What is more, downloading of large raster datasets is a quite time-consuming process.

Thus, in the frame of this work, we focus on size-based analysis. We disclosed that size data could be easily obtained from the headers, which are available for all tiles and are accessible by special requests. We conducted massive requests of raster tiles headers to
collect size information of Google Tiles. OSM data were rendered locally using standard OSM style because the OSM project provides all required data and tools. Then, we have compared OSM and GM tiles sizes. Differences of OSM and GM file sizes are used as a data quality indicator in a tile. Bigger size indicates higher quality. Images in the PNG format are examined for this.

In addition to the size analysis, for OSM raster tile data, we extracted edges. As mentioned, OSM raster tiles have been preliminarily rendered to prevent massive downloading of large raster data. The extracted edges are processes as a set of points. A point is a pixel (X and Y pixel coordinates) of a detected edge. An overall number of points (pixels) reflect the quantity of information provided by a tile.

Figure 3 demonstrates a sample result of the Canny edge detection for a tile; tile coordinates and file size are provided. We use the following command of the ImageMagick package (Still, 2006) for caring out the algorithm: convert infile.png -color-space gray -canny 0 × 4 + 5%+5% outfile.png.

4.4. Simplified data quality indicator

Rousell et al. (2018) have proposed a Simplified Data Quality Indicator (SDQI). SDQI is a data quality indicator aggregating various parameters. For this, a set of parameters are defined. Each parameter represents an essential aspect of data or metadata. Moreover, a lower value indicates lower fitness-of-use and vice versa. The following parameters are required: the number of points, number of lines, length of lines, number of polygons, length of polygons boundaries, areas of polygons, and number of attributes. These parameters are applicable to all types of spatial vector datasets. Moreover, additional parameters are calculated for OSM data: number of tile hits (represents an OSM tile popularity; it is delivered by the OSM planet portal), number of contributors, number of changesets, average version, average timestamp. SDQI is calculated as follows:

(a) Original OSM tile (X=80440, Y=87302, ZoomLevel=19, file size in bytes 187302).
(b) The corresponding Canny edge detection results.

Figure 3. Raster tile edge detection.
In the equation, $i$ is a parameters index (the full list of parameters is provided above); $\min v_i$ and $\max v_i$ are minimal and maximal values of a parameter, correspondingly. $\min c$ and $\max c$ are minimal and maximal class numbers. A class number in other words can be considered as quality level. The equation splits data into 11 classes according to the data quality. A lower value indicates lower data quality and vice versa. SDQI is calculated for OSM tile rectangle areas in zoom level 19. In addition to the mentioned parameters, in this work, we introduce one new source. The Canny edge number of points (see 4.3 for details) for each tile region of OSM is applied in addition to the other parameters.

4.5. Conformance quality level

As proposed at the beginning of the section, two rules-of-thumb are defined. The rules are based on an intensive visual and statistical analysis of data and the results of the proposed methods. In Section 5, we formulate a rule to EDP statistics results. The rule allows us to preliminary filter out the low-quality OnToMap data layers. SDQI and size-based statistics are aggregated on bar-charts diagrams to demonstrate the second rule-of-thumb. For the rule definition, we also use a list of pilot sites ordered according to the data-type analysis results. The rule establishes conformance quality (SDQI) level (i.e. minimal accepted data quality level); only data with quality higher or equal an accepted levels are included into the proposed embeddable services. For this, we use SDQI classes.

The prs/IGIStk/q/wbi.tcl script implements the proposed data quality indicator. It provides functions for calculating the required parameters and SDQI. Results are stored in a GSDR database instance. The script intensively utilizes the prs/tilescf/libtiles.tcl library of the archiving software for tile-based processing.

5. Results

Embeddable systems comprising of the prepared databases according to the proposed methodology (WeGovNowConsortium & OpenStreetMapContributors, 2019) and the corresponding software (Noskov, 2019a) are the results of this work. The archiving datasets contain the source and resulting datasets. The archiving software consists of the applications and libraries amalgamation (Tiles.CF, IGIS.TK, additional tools, and libraries) developed for the preparation of the embeddable databases and a part of the source code of GSDR.GQ required for the deployment of the embeddable systems.

5.1. Resulting data model and datasets

Table 3 shows a list of the resulting files. The *.sqlite files are SpatiaLite GSDR databases. *.csv.gz are comma-separated value compressed text files. They mainly provide content dumps of the full GSDR database’s tables. *.linestats.txt are line statistics resulting files.
linestats.HTML provides the line statistics in human-readable HTML format. *.poly provides boundaries of pilot sites suitable for various OSM tools. *.osm.bz2 are OSM data clipped from the OSM full history dump. *.json.gz are compressed GeoJSON files providing masks utilized for the extraction of OSM (sdqimask.json.gz), and OnToMap’s PSI (sdqipsimask.json.gz) data for GSDR pilot sites’ instances. ontomapdata.tar.gz archives data layers provided by OnToMap. opendata_sandona.zip, opendata_southwark.rar, and opendata_turin.rar comprises original (without modifications) Open Data in the ESRI shape-file format.

Many of the described files have been dumped from the central GSDR database (qwgnsdata.sqlite.gz). All *.sqlite.gz files utilize a data model introduced in Figure 4. The presented data model has been derived from the main GSDR database. Pilot site instances of the database comprise only the following tables: elements, tags, keys, vals, tiles and eltile.

In Figure 4, three types of data are presented: master, tile index, quality assessment and supplementary information. Master data are broken down into the four following tables. First, the elements (the elements.csv.gz corresponding data dump) table provides

| # | File name                        | Description                                                                 | File size  |
|---|----------------------------------|-----------------------------------------------------------------------------|------------|
| 1 | corpts.csv.gz                    | PSI and Open Data with correspondences                                       | 40.1 Mb    |
| 2 | edpotmdist.csv.gz                | Point data with distances from PSI to OSM                                    | 6.8 Mb     |
| 3 | elements.csv.gz                  | See Figure 4                                                                 | 25.1 Mb    |
| 4 | eltile.csv.gz                    | See Figure 4                                                                 | 5.1 Mb     |
| 5 | gsdr_wgn_sd.sqlite               | SD embeddable database                                                       | 22.3 Mb    |
| 6 | gsdr_wgn_sw.sqlite               | SW embeddable database                                                       | 119.8 Mb   |
| 7 | gsdr_wgn_tr.sqlite               | TR embeddable database                                                       | 71.2 Mb    |
| 8 | hd.linestats.txt                 | HD OSM XML raw line statistics                                              | 428 B      |
| 9 | hd.osm.bz2                       | Clipped OSM data for SD                                                     | 12 Mb      |
| 10| hd.poly                          | HD boundary                                                                  | 145 B      |
| 11| keys.csv.gz                      | See Figure 4                                                                 | 20 Kb      |
| 12| linestats.HTML                    | Line statistics visualization                                                | 11 Kb      |
| 13| nocsorpts.csv.gz                 | PSI and Open Data without correspondences                                    | 3.1 Mb     |
| 14| ontomapdata.tar.gz               | PSI data extracted from OnToMap                                              | 8.7 Mb     |
| 15| opendata_sandona.zip             | SD original open data                                                        | 357 Kb     |
| 16| opendata_southwark.rar           | SW original open data                                                        | 4.9 Mb     |
| 17| opendata_turin.rar               | TR original open data                                                        | 4.1 Mb     |
| 18| osmraspt_crptsnum.csv.gz         | Number of Canny edge point extracted from OSM tiles                          | 1.6 Mb     |
| 19| osmrastpts.csv.gz                | Canny edge point extracted from OSM tiles                                    | 2.4 Gb     |
| 20| qwgnsdata.sqlite.gz              | Full database                                                                | 836.3 Mb   |
| 21| sandona.poly                     | Clipped OSM data for SD                                                      | 2 Kb       |
| 22| sd.linestats.txt                 | SD OSM XML raw line statistics                                              | 418 B      |
| 23| sd.osm.bz2                       | SD boundary                                                                  | 2.2 Mb     |
| 24| sdqidata.csv.gz                  | See Figure 4                                                                 | 5.8 Mb     |
| 25| sdqimask.json.gz                 | A mask for OSM data                                                         | 132 Kb     |
| 26| sdqipsimask.json.gz              | A mask for PSI data                                                         | 99 Kb      |
| 27| southwark.poly                   | Clipped OSM data for SW                                                      | 150 B      |
| 28| sw.linestats.txt                 | SW OSM XML raw line statistics                                              | 430 B      |
| 29| sw.osm.bz2                       | Clipped OSM data for SW                                                     | 12.1 Mb    |
| 30| tags.csv.gz                      | See Figure 4                                                                 | 16.6 Mb    |
| 31| tiles.csv.gz                     | See Figure 4                                                                 | 4.2 Mb     |
| 32| tilesizes.csv.gz                 | Tile sizes                                                                   | 2.9 Mb     |
| 33| tr.linestats.txt                 | TR OSM XML raw line statistics                                              | 426 B      |
| 34| tr.osm.bz2                       | TR OSM data                                                                  | 6 Mb       |
| 35| turin.poly                       | TR boundary                                                                  | 2 Kb       |
| 36| vals.csv.gz                      | See Figure 4                                                                 | 2.7 Mb     |
| 37| wgntilelog.sqlite.gz             | OSM tile-hits database                                                      | 460.1 Mb   |
the primary information regarding a geographic data feature. All features have the id (primary key), dataid (an id derived from a source dataset), version (a version of a feature), uid (user id), timestamp (object creating time), changeset, datasrc and geom. The datasrc column has one of the following values: “OSM.CUR.HD,” “OSM.CUR.SD,” “OSM.CUR.SW,” “OSM.CUR.TR,” “PSI.ODT.SD,” “PSI.ODT.SW,” “PSI.ODT.TR,” “PSI.OTM.SD,” “PSI.OTM.SW” or “PSI.OTM.TR”. geom stores geometric features. Through tags (tags.csv.gz) every element feature is joined with a corresponding keys (keys.csv.gz) and vals (vals.csv.gz) text (txt columns) entities. The master data tables are created according to the following SQL code (see IGIStk/c/geojson2spalte.tcl of the archiving software for more information):

```sql
CREATE TABLE elements (
id INTEGER PRIMARY KEY,
dataid TEXT,
version INTEGER,
type INTEGER,
uid INTEGER,
timestamp INTEGER,
changeset INTEGER,
datasrc TEXT);
SELECT AddGeometryColumn('elements', 'geom', 4326, 'GEOMETRY', 'XY');
CREATE TABLE keys (txt TEXT,
UNIQUE (txt) ON CONFLICT IGNORE);
CREATE TABLE vals (txt TEXT,
UNIQUE (txt) ON CONFLICT IGNORE);
CREATE TABLE tags (id INTEGER,
key INTEGER,
val INTEGER,
PRIMARY KEY (id, key, val) ON CONFLICT IGNORE,
FOREIGN KEY (id) REFERENCES elements(id),
```

Figure 4. Data model.
The second data type is a tile index consisting of the tiles (tiles.csv.gz) and eltile (eltile.csv.gz) tables. The former includes five tables: q, col, row, geomcent and geompol. q is a quadkey value. Schwartz (2018) has proposed a quadkey implementation and described its special properties, which make quakeys extremely useful for the spatial indexing and data aggregation. We intensively use these properties in the frame of the project. In addition to the quadkey, we take advantages of the normal tile coordinates stored in col ("X" coordinate) and row ("Y" coordinate). All tiles are in zoom level 19; thus, we do not have a zoom level information in the database.

Moreover, the table manages geomcent and geompol columns. They handle geometries of tile centroid and polygon objects. eltile is a tile-based spatial index joining the elements and tiles tables.

The sdqidata (sdqidata.csv.gz) table provides the data quality assessment results and related parameters. Aggregated results are resided in the sdqi column. The name column concludes on of the following values: “OSM.CUR.SW,” “OSM.CUR.TR,” “OSM.CUR.HD,” “OSM.CUR.SD,” “OSM.ODT.SW,” “OSM.ODT.TR,” “OSM.ODT.HD,” “PSI.OSM.SW,” “PSI.OSM.TR,” “PSI.OSM.HD” and “PSI.OTM.HD.” In addition to the datasrc column of the elements table, it utilizes “PSI.OSM.*” data layers. The meaning of such values is PSI derived from OSM data. The selected OSM features correspond the OnToMap data types. The “highway” data have been excluded from the SDQI processing; Section 5.2 discusses it.

Finally, the cannypts (osmrast_ptsnum.csv.gz), edpotmdist (edpotmdist.csv.gz), psiedp (edpotmdist.csv.gz) and ps. cannypts stores edge information (points number in the count column) extracted from OSM raster tiles joined with tiles through the q (quad key) column. ps collects polygons of the pilot sites.

5.2. Equidistant points statistics

EDP results enable us to evaluate the OnToMap. For this, we compare OnToMap data with source Open Data and, then, with OSM. The OnToMap and Open Data comparison are based on 1-m EDP. The psiedp table of the full database provides the prepared EDP data. The table contains point integer coordinates in the UTM projection. For each coordinate pair, we calculated the number of points belongs to a pair. If both data types (PSI OnToMap and Open Data) are linked to a coordinate pair, this pair is marked as correct and, thus, appeared in corpts.csv.gz. Otherwise, it is considered as incorrect and stored in nocorpts.csv.gz.

Several inferences can be concluded from the resulting EDP data. First, one can notice that Southwark EDP comprise very few points (52) without correspondences. That indicates that almost all data features from the source Open Data provided by the municipalities appear almost unchanged in the PSI OnToMap database. Second, the San Dona di Piave EDP datasets comprise up to 160 points without correspondences (points appeared in OnToMap database only). After the examination of these points, we have concluded that the source data set we have is not complete. Few data sets are not included in the Open Data layers transferred to us. Third, the Turin EDP data have massive points without correspondences; both single Open Data and OnToMap points are marked. It means that
many data features were ruled out during the data selecting. Moreover, there are many OnToMap points without Open Data pairs. It indicates that several source Open Data layers are not included in the dataset provided for the evaluation.

In Figure 5, sample points without correspondences are depicted. The number of points with and without pairs are as follows (number of pair points, without correspondences, ratio, PSI OnToMap single points, and Open Data single points):

- Southwark: 1,250,815, 0.004%, 52, 26, 26;
- San Dona di Piave: 67,834, 160, 0.2%, 160, 0;
- Turin: 1,281,459, 185,054, 14.4%, 116,627, 68,427.

From the provided statistics, one can conclude that the Southwark Open Data are complete and corresponded to the OnToMap database. For San Dona di Piave, lacking source datasets should be found. For Turin data, many layers are derived from unknown data sources. The original data does not contain about 5.3% of the data presented in OnToMap.

In addition to the described earlier analysis, EDP data are utilized for the evaluation and filtering out of the OnToMap data against OSM features. According to 4.1, the shortest distances from the 3-m OnToMap EDP to the 1-m OSM EDP are calculated (see the edpotmdist.csv.gz resulting file). The results are presented in Table 4 according the pilot sites and layer names. A complete list of the OnToMap’s layer names is as follows:

![Figure 5](image-url). One-meter EDP without correspondences: OnToMap (blue point), Open Data (red points). Grey – OnToMap data; red lines – boundaries of pilot sites. Left – Turin (Parco Data in the center). Right – San Dona di Piave.
Table 4. Shortest distances from three-meter OnToMap EDP to one-meter osm EDP. Values bigger than the values defined by the rule-of-thumb are marked by a bold font.

| Data                        | Number | Min | Max | Avg | Median | Stdev |
|-----------------------------|--------|-----|-----|-----|--------|-------|
| PSI.OTM.SD.ArtGallery       | 1      | 3.7 | 3.7 | 3.7 | 3.7    | 0     |
| PSI.OTM.SD.BicyclePath      | 22,327 | 0   | 29.9| 2.6 | 1.9    | 3     |
| PSI.OTM.SD.Business         | 724    | 0   | 29.8| 3.1 | 2.2    | 3.5   |
| PSI.OTM.SD.Cafe             | 85     | 0   | 15.1| 2.9 | 2.2    | 2.9   |
| PSI.OTM.SD.Cinema           | 3      | 3.1 | 5.2 | 4.4 | 4.8    | 1.1   |
| PSI.OTM.SD.Club             | 20     | 0.4 | 7.6 | 3.7 | 3.5    | 2.2   |
| PSI.OTM.SD.CulturalCenter   | 1      | 1.3 | 1.3 | 1.3 | 1.3    | 0     |
| PSI.OTM.SD.FinancialService | 175    | 0   | 15.6| 2.8 | 2.1    | 2.3   |
| PSI.OTM.SD.FoodAndAccom     | 181    | 0.1 | 19  | 2.8 | 2.1    | 2.8   |
| PSI.OTM.SD.HealthSocialService | 94   | 0.3 | 17  | 4.4 | 3.8    | 3     |
| PSI.OTM.SD.InformationCom   | 107    | 0.2 | 29  | 4.2 | 2.6    | 4.6   |
| PSI.OTM.SD.Leisure          | 63     | 0.1 | 1.7 | 3.3 | 2.8    | 2.3   |
| PSI.OTM.SD.Library          | 1      | 7.1 | 7.1 | 7.1 | 7.1    | 0     |
| PSI.OTM.SD.Logistics        | 67     | 0.2 | 12.2| 3.2 | 2.7    | 2.2   |
| PSI.OTM.SD.Manufacturing   | 262    | 0.2 | 24.6| 4.2 | 3.2    | 3.5   |
| PSI.OTM.SD.Monument         | 23     | 0.3 | 16.3| 6.3 | 5.9    | 4.2   |
| PSI.OTM.SD.ProfessionalScienti   | 191  | 0.2 | 29.8| 4   | 2.5    | 5.1   |
| PSI.OTM.SD.RealEstateActivity | 996  | 0   | 29.8| 3.8 | 2.8    | 4.1   |
| PSI.OTM.SD.Restaurant       | 58     | 0.2 | 17.1| 2.7 | 2.1    | 2.6   |
| PSI.OTM.SD.School           | 65     | 0.2 | 11.2| 4   | 3.7    | 2.7   |
| PSI.OTM.SD.Store            | 1110   | 0.1 | 29.8| 3.4 | 2.5    | 3.4   |
| PSI.OTM.SD.WaterAndWasteDis | 9      | 0.7 | 29.8| 6.7 | 4.3    | 8.9   |
| PSI.OTM.SW.BicyclePath      | 1014   | 0   | 16.5| 2.3 | 1.8    | 2.1   |
| PSI.OTM.SW.Highway          | 286,418| 0   | 29.9| 4.1 | 2.4    | 4.9   |
| PSI.OTM.SW.HistoricalCenter | 36,191 | 0   | 29.9| 3.5 | 2      | 4.4   |
| PSI.OTM.SW.ParkingLot       | 80,533 | 0   | 12.5| 2.1 | 1.7    | 1.6   |
| PSI.OTM.SW.School           | 390    | 0.1 | 26.6| 7.3 | 5.3    | 6     |
| PSI.OTM.SW.TrainStation     | 14     | 0.4 | 8.8 | 2.6 | 1.3    | 2.7   |
| PSI.OTM.SW.UrbanPark        | 26,496 | 0   | 29.7| 2.4 | 1.7    | 2.6   |
| PSI.OTM.TR.Cinema           | 61     | 0.1 | 5.5 | 2.3 | 2.2    | 1.3   |
| PSI.OTM.TR.Drugstore        | 225    | 0.1 | 9.6 | 1.2 | 1      | 1     |
| PSI.OTM.TR.Hospital         | 6468   | 0   | 29.9| 4   | 1.9    | 5.3   |
| PSI.OTM.TR.LawEnforcement   | 66     | 0.1 | 6.1 | 1.4 | 1.1    | 1.3   |
| PSI.OTM.TR.PlaceOfWorship   | 8443   | 0   | 29.9| 3.7 | 1.7    | 5     |
| PSI.OTM.TR.School           | 61,099 | 0   | 29.9| 4.2 | 1.8    | 6     |
| PSI.OTM.TR.StreetMarket     | 7123   | 0   | 29.1| 2.4 | 1.7    | 2.4   |
| PSI.OTM.TR.UrbanPark        | 421,023| 0   | 29.9| 3.9 | 2.7    | 4.2   |

- San Dona di Piave (25 layers): Accommodation, ArtGallery, BicyclePath, Business, Cafe, Cinema, Club, CulturalCenter, FinancialService, FoodAndAccommodation, HealthSocialService, InformationCommunicationService, Leisure, Library, Logistics, Manufacturing, Monument, Museum, PowerDistribution, Professional ScientificTechnicalActivity, RealEstateActivity, Restaurant, School, Store, WaterAndWasteDisposal;
- Turin: Cinema, Drugstore, HealthSocialService, Hospital, LawEnforcement, Leisure, PlaceOfWorship, School, StreetMarket, UrbanPark;
- Southwark: BicyclePath, Highway, HistoricalCenter, Monument, ParkingLot, PlaceOfWorship, School, TrainStation, UrbanPark.

We have carried out a detailed visual and statistical analysis of the datasets and defined the first rule of thumb. The rule says that layers comprise more than 500 points with average, median and standard deviation more than 4, 2 and 4, correspondingly,
should be excluded from the further processing, because they are inconsistent with the OSM features. In Table 4, values meeting the criteria are marked by a bold font. According to the table, the “PSI.OTM.SW.Highway” layer should be excluded. It is recommended to exclude the layer from the WGN platform completely. OSM highway data provided by GSDR.GQ can be used instead.

5.3. Data-type evaluation results

Table 5 provides the line statistics results (see also the archived *linestats.txt row data files and linestats.HTML aggregated data document). First of all, one can conclude that there are no “negative” (“-“-suffixed) classes. All occurrences of all such classes equal “0”. That means that all the examined OSM data dumps are consistent and do not have problems and imperfections on the data format and XML structure level. In the table, scores “1,” “2,” “3” and “4” are marked by regular, italic, bold and bold-italic fonts, correspondingly. Scores are given using the normalized lines number values.

According to the resulting scores, the pilot sites are ordered from the lower to higher quality as follows: San Dona di Piave (19), Turin (39), Heidelberg (55) and Southwark (67). Further, we define conformance quality levels for the OSM data for each pilot site. The second rule-of-thumb is as follows: a minimal acceptable SDQI quality level is more or equal the SDQI quality level of a pilot site with fewer scores according to the data-type model.

5.4. Simplified data quality indicator calculation and raster tile processing results

SDQI is calculated for every tile of the pilot sites. SDQI of OSM data is provided for all pilot sites. While, SDQI of PSI data is delivered for three pilot sites (excluding Heidelberg), which are the WGN’s pilot sites. PSI’s SDQI does not cover OSM-specific parameters (number of hits, contributors, changesets, canny points, average version, and timestamps). It is calculated for the comparison of the PSI sources: OSM, OnToMap and Open Data.

As mentioned in 4.3, files sizes of PNG raster tiled web maps of OSM and GM are compared. File sizes were obtained without actual downloading of PNG files; they have been retrieved from headers provided by a web server. In general, bigger tile size means that it provides a more quantity of information.

As mentioned earlier, the Canny edge detection requires downloading raster tiles. This is quite problematic because map tile providers restrict access and prevent massive downloading. Thus, for all tile maps, excepting OSM, only tile sizes have been derived. We rendered all required OSM tiles to conduct the Canny edge processing. This enables to verify if sizes can actually represent quantity of information. The applied Canny edge method ignores noise (e.g. polygonal patterns). Therefore, strong correlation of Canny edge points and tile sizes proves the applicability of tile sizes. Table 6 confirms high linear correlation of the number of Canny edge points and tile sizes.

According to the harvested size statistics, empty tiles (i.e. tiles filled by only one color) have the following sizes: OSM – 81, GM – 215. Further, we use the difference of these values as a “gap” value (215–81 = 134). In order to prevent the influence of the PNG
| Attribute name | Heidelberg        | San Dona          | Southwark         | Turin             |
|----------------|-------------------|-------------------|-------------------|-------------------|
| lines          | 2,530,197         | 458,375           | 2,661,966         | 1,473,814         |
| chars          | 165,988,299 (1.126E+02) | 34,919,699 (2.369E+01) | 168,024,430 (1.140E+02) | 94,605,273 (6.419E+01) |
| sblank         | 8,263,740 (5.607E+00) | 1,413,536 (9.591E-01) | 8,810,992 (5.978E+00) | 4,729,122 (3.209E+00) |
| ats            | 90,364,243 (6.131E+01) | 19,365,896 (1.314E+01) | 93,092,114 (6.316E+01) | 53,381,869 (3.622E+01) |
| noatrs         | 67,360,316 (4.570E+01) | 14,140,267 (9.594E+00) | 66,121,324 (4.486E+01) | 36,494,282 (2.476E+01) |
| stags          | 9,680,287 (6.568E+00) | 1,697,083 (1.151E+00) | 10,096,675 (6.851E+00) | 5,327,011 (3.614E+00) |
| ctags          | 317,670 (2.155E-01)     | 49,580 (3.364E-02)     | 1414,824 (2.813E-01)     | 360,556 (2.446E-01)     |
| slashmore      | 4,425,056 (3.002E+00) | 817,592 (5.547E-01) | 4,494,286 (3.049E+00) | 2,226,518 (1.511E+00) |
| more           | 317,669 (2.155E-01)     | 49,579 (3.364E-02)     | 414,823 (2.813E-01)     | 360,555 (2.446E-01)     |
| mblanks1       | 8,429,068 (5.719E+00) | 1,724,456 (1.170E+00) | 8,335,417 (5.656E+00) | 4,623,310 (3.137E+00) |
| noatrsaz       | 35,761,498 (2.426E+01) | 8,077,521 (5.481E+00) | 34,029,882 (2.309E+01) | 18,973,022 (1.287E+01) |
| dquotes        | 16,858,136 (1.144E+01) | 3,448,912 (2.340E+00) | 16,670,834 (1.131E+01) | 9,246,620 (6.274E+00) |
| atrsblanks     | 191,450 (1.299E-01)     | 19,724 (1.338E-02)     | 1,132,696 (7.685E-01) | 343,010 (2.327E-01) |
| atrs09         | 50,849,667 (3.450E+01) | 12,046,340 (8.174E+00) | 46,094,568 (3.128E+01) | 26,494,417 (1.798E+01) |
| atrsaz         | 14,770,431 (1.002E+01) | 2,362,232 (1.603E+00) | 20,745,112 (1.408E+01) | 12,315,721 (8.356E+00) |
| atrsAZ         | 2,409,364 (1.635E+00) | 402,052 (2.728E-01) | 3,294,365 (2.235E+00) | 1,892,739 (1.284E+00) |
| atrspunct      | 5,280,787 (3.583E+00) | 1,084,071 (7.356E-01) | 5,151,912 (3.496E+00) | 3,088,311 (2.095E+00) |
| atrsgraph      | 4,408 (2.991E-00)     | 2,565 (1.740E-03) | 2,627 (1.782E-03) | 1,051 (7.131E-04) |
| SCORE          | 55                  | 19                | 67                | 39                |
encoding model chosen by tile providers, we normalize the OSM – GoogleMaps tile size differences using the gap. Tiles’ SDQI groups the resulting statistics of the file-size differences.

In order to define conformance quality level of OSM data, we analyze tile file size according to the SDQI classes. Figure 6 illustrates the results. In the figure, the following parameters are depicted (Y-axis): standard deviation (dark red bars), minimal (red), median (green), average (blue) and maximal (yellow) values of OSM-GoogleMaps file-size differences. All parameters are calculated using the following equation: \((\text{osm\_size} - \text{googlemaps\_size})/\text{gap}\). Moreover, minimal and maximal values are divided by 10.

Using the bar charts, we have specified the third rule-of-thumb (the first and second rules are introduced in two previous subsections): a minimally acceptable data quality level is bigger than SDQI class with either median or average value more than two of standard deviation, minimal or maximal values, while these two values are not equal zero. According to Figure 6 and two rules-of-thumb, an acceptable data quality levels for pilot sites are as follows: (a) San Dona di Piave – 4, (b) Turin – 4, (c) Heidelberg – 2, (d) Southwark – 2. For San Dona di Piave, “4” has been defined, because in class “3” median and average values are lower than all other values. For Turin, in class “4”, median and average values are lower than standard deviation and maximal. Thus, according to the third rule-of-thumb, the minimally acceptable quality level should be “5”, but it is “4” because of the second rule-of-thumb (i.e. it should be less or equal the level of a pilot site with lower OSM data-type model quality). For Heidelberg, the minimally acceptable quality level is “2”, because in the class “1,” a minimal value equals zero. For Southwark, the minimally acceptable quality is “3” according to the third rule-of-thumb, but it is defined as “2” because of the second rule-of-thumb.

A final rule-of-thumb allows defining minimally acceptable SDQI level for PSI data using the OSM, PSI OnToMap and PSI OSM SDQI results. The rule says that PSI OnToMap SDQI should be more than PSI OSM SDQI for tiles with OSM SDQI more or equal a minimally acceptable data quality level. Moreover, PSI OnToMap SDQI level should have “+2” class in comparison with PSI OSM SDQI for tiles with OSM SDQI less a minimally acceptable data quality level. The rule can be formulated as follows:

\[
Q_{\text{psiotm}} - Q_{\text{psiosm}} \geq -1, \quad \text{when } Q_{\text{curosm}} \geq Q_{\text{minosm}},
\]

\[
Q_{\text{psiotm}} - Q_{\text{psiosm}} > 2, \quad \text{when } Q_{\text{curosm}} Q_{\text{minosm}}.
\]

Tiles corresponding to the described rule are marked in Figure 7 as an OTM mask. The masks depict areas used for the selection PSI OnToMap data for the resulting databases.
Figure 6. File-size statistics (Y-axis) according to the SDQI classes (X-axis). a) San Dona di Piave, b) Turin, c) Heidelberg, d) Southwark.
of the pilot sites. In the figure, OSM SDQI classes are displayed by color background. For each pilot site, the third rule-of-thumb defines, which specifies masks for the selection of OSM data masks. One can notice that significant area of the San Dona di Piave and Turin sites have SDQI level below a minimally acceptable quality level. While, almost whole areas of Heidelberg and Southwark pilot sites are covered by tiles with SDQI level greater than minimally acceptable quality levels.

The results are utilized by embeddable instances of GSDR. A database of such instance comprises only data having quality equal of higher the minimal acceptable quality defined

Figure 7. SDQI results. Color background is OSM SDQI levels. Hatching areas are masks used for the OnToMap data selection.
earlier. Thus, low-quality data are excluded from the resulting databases. Thus, embeddable systems rely on features with the ensured quality. This significantly reduces the size of a resulting database making it highly portable and increases the trustfulness of a system in general.

Embeddable systems can be deployed on any unix server. Figure 8 shows various uses of a running system. Web application can utilize the offered functionality for users input improvement. For instance, instead of creating new spatial objects users could pick up existing object from an embeddable database. Moreover, spatial snapping functionality improves the precision of users spatial contributions. Finally, text autocompleting and spell-checking functionality is offered. Since the provided API is quite flexible, additional facilities can be implemented using the prepared solution. In Appendix, concrete instructions for deployment of embeddable systems are provided.

6. Conclusion

In this work, we introduced a concept and implementation of embeddable web services for data quality assurance designed for WebGIS open-data-driven applications. The proposed solutions are applicable for modern open-data web services, especially non-commerce instances.
The software facilitates two main tasks: the data preparation and embeddable web-service functionality. The data preparation breaks down into the following steps of the EDPs analysis, applying of the data-type model and definition of conformance quality levels for pilot sites. EDP statistics allows preliminary evaluation and filtering out public-sector-information data layers. The data model enables to order pilot sites according to the OSM data quality. The definition of conformance quality levels is based on the comparison of OSM and GM map tiles and OSM Simplified Data Quality Indicator.

The conducted quality assessment allows specifying areas with a satisfactory quality of OSM data. These data are included in embeddable databases. Moreover, we use SDQI to select appropriate public-sector-information Open Data features for embeddable databases.

The second type of the archiving software is required for the functionality of embeddable instances of Geo-Spatial Data Repository. We have demonstrated the concrete steps required for the installation of an instance as a part of Apache Web Server. Installed instances can be utilized as shared services for quality assurance and improved. We demonstrated a number of use cases: text completing, spell-checking, object-picking and snapping. Moreover, generated quality assessment resulting data provide an inventory of examined geospatial datasets.

The described solutions are delivered as an open-source and open-data system. It consists of a number of software projects easily available online and various datasets. Both are archived in a general-purpose open-access repository. This makes it possible to use our solutions in broad research and development areas.

Acknowledgments

This work has been funded by the European Union’s Horizon 2020 research and innovation program under the grant agreement n. 693514 (“WeGovNow”). The article reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

Disclosure statement

No potential conflict of interest was reported by the authors.

Data availability statement

The data that support the findings of this study are available at https://doi.org/10.5281/zenodo.2454150.

Funding

This work was supported by the H2020 European Research Council [693514 WeGovNow].

ORCID

Alexey Noskov http://orcid.org/0000-0003-0503-6558
Alexander Zipf http://orcid.org/0000-0003-4916-9838
**Glossary**

**GeoJSON** is a format for encoding a variety of geographic data structures extending JSON.

**geojsplit** is an application for splitting GeoJSON FeatureCollections to smaller files (https://github.com/woodb/geojsplit).

**GNU Make** is a build automation tool that automatically builds executable programs and libraries from source code by reading files called Makefiles which specify how to derive the target program.

**GSDR** Geo-Spatial Data Repository for "Grand" Quality, is a web service for spatial data quality assessment (https://gsdr.gq).

**HTML5** is the latest evolution of the HTML standard providing a rich facilities for development of rich-functionality web pages.

**IGIS.TK** Integrated Geographic Information System Tool Kit, is an open source software framework collecting a number of tools and libraries for geospatial data quality assessment, integration and conversion (http://igis.tk).

**JSON** JavaScript Object Notation, is a lightweight data-interchange format.

**Linux** is a family of free and open-source software operating systems based on the Linux kernel.

**OGR** is a C++ open source library (and commandline tools) providing read (and sometimes write) access to a variety of vector file formats.

**OSM** OpenStreetMap, is a collaborative free editable map of the world.

**osm2geojson** is an application for streaming OSM data in XML format to GeoJSON (https://github.com/rclark/osm2geojson).

**RestAPI** is a modern method which allows communication between a web-based client and server.

**Tcl** Tool Command Language is a powerful easy dynamic programming language, suitable for a very wide range of uses, including web and desktop applications, networking, administration, testing etc. (https://tcl.tk).

**Tiles.CF** Tiles Common Framework, is a collection of open source libraries for geotiles and equidistant-points analysis (http://tiles.cf).

**Unix** is a family of multitasking, multiuser computer operating systems.

**WebGIS** Map-based web applications providing advanced spatial-oriented functionality.

**XML** is a markup language much like HTML. XML was designed to store and transport data.

**References**

Alenezi, H., Tarhini, A., & Masadeh, R. (2015). Investigating the strategic relationship between information quality and e-government benefits: A literature review. *International Review of Social Sciences and Humanities, 9*(1), 33–50. Retrieved from https://bura.brunel.ac.uk/bitstream/2438/10963/1/FullText.pdf

Al-Masri, E., & Mahmoud, Q. H. (2007, August). Qos-based discovery and ranking of web services. In *2007 16th international conference on computer communications and networks* (p. 529–534). doi:10.1109/ICCCN.2007.4317873

Antonini, A., Boella, G., Calafiore, A., Salaroglio, C., Sanasi, L., & Schifanella, C. (2016). First life, the neighborhood social network: A collaborative environment for citizens. *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion - CSCW 16 Companion*. doi:10.1145/2818052.2874310

Ardissonno, L., Lucenteforte, M., Mauro, N., Savoca, A., Voghera, A., & La Riccia, L. (2017). Ontomap: Semantic community maps for knowledge sharing. In *Proceedings of the 28th acm conference on hypertext and social media* (pp. 317–318). New York, NY, USA: ACM. doi:10.1145/3078714.3078747

Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009, July). Methodologies for data quality assessment and improvement. *ACM Computing Surveys, 41*(3), 16:1–16:52.

Belongie, S., Malik, J., & Puzicha, J. (2002, April). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(4), 509–522.
Bertot, J. C., Gorham, U., Jaeger, P. T., Sarin, L. C., & Choi, H. (2014). Big data, open government and e-government: Issues, policies and recommendations. *Information Polity, 19*(1–2), 5–16.

Boella, G., Francis, L., Grassi, E., Kistner, A., Nitsche, A., Noskov, A., & Tsampoulidis. (2018, April). WeGovNow: A map based platform to engage the local civic society. In *Proceedings of the 8th international workshop on location and the web - locweb’18*. ACM. doi:10.1145/3184558.3191560

Chan, H., Chieu, T., & Kwok, T. (2008, September). Autonomic ranking and selection of web services by using single value decomposition technique. In *2008 ieee international conference on web services* (p. 661–666). doi:10.1109/ICWS.2008.124

Doytsher, Y., & Noskov, A. (2015, February). A segmentation-based approach for improving the accuracy of polygon data. In *Geoprocessing 2015: The 7th intentional conference in advanced geographic information systems, applications, and services* (Vol. 1, pp. 69–74). Lisbon, Portugal: IARIA.

Flynt, C. (2012). *Tcl/tk. a developer’s guide*. Morgan Kaufmann. Retrieved from https://www.elsevier.com/books/tcl-tk/flynt/978-0-12-384717-1

Furieri, A. (2019). *SpatiaLite* [Online]. Retrieved from https://www.gaia-gis.it/fossil/libspatialite/index

Haldar, S. (2016). *Sqlite database system design and implementation (version 2)*. Sibsankar Haldar - Self Publishing.

KöRner, P. (2017). *OSM-history-splitter* [Online]. Retrieved from https://github.com/MaZderMind/osm-history-splitter

Loshin, D. (2001). *Enterprise knowledge management: The data quality approach*. San Francisco, CA: Morgan Kaufmann.

Ma, T., & Latecki, L. J. (2011, June). From partial shape matching through local deformation to robust global shape similarity for object detection. In *Cvpr 2011* (p. 1441–1448). doi:10.1109/CVPR.2011.5995591

Masse, M. (2011). *Rest api design rulebook: Designing consistent restful web service interfaces*. Sebastopol, CA: O’Reilly Media, Inc.

Noskov, A. (2018a, June). Computer vision approaches for big geo-spatial data: Quality assessment of raster tiled web maps for smart city solutions. In T. Bandrova & M. Konecny (Eds.), *Proceedings 7th international conference on cartography and gis* (Vol. 1, pp. 296–305). Sozopol, Bulgaria: Bulgarian Cartographic Association.

Noskov, A. (2018b, August). Open source tools for coastal dynamics monitoring. In *Sixth international conference on remote sensing and geoinformation of the environment (rscy2018)* (Vol. 107731C). SPIE. doi:10.1117/12.2326277

Noskov, A. (2019a). * Archived tools: The amalgamation of the GSDR.GQ, Tiles.CF and N-kov Libs and Tools*. doi:10.5281/zenodo.2455025

Noskov, A. (2019b). *Collection of general-purpose tools and libraries* [Online]. Retrieved from http://n-kov.com/LibsAndTools/

Noskov, A. (2019c). *Integrated geographic information system tool kit: A collection of highlevel tools for spatial data conversion, fusion, generalization and quality management* [Online]. Retrieved from http://igis.tk or http://igis.n-kov.com

Noskov, A. (2019d). *Tiles common framework: A library for web tiled maps* [Online]. Retrieved from http://tiles.cf or http://tiles.n-kov.com

Noskov, A., Grinberger, A. Y., Papapesios, N., Rousell, A., Troilo, R., & Zipf, A. (2019). Data-type model for low-level inventory of big data files: A case study of openstreetmap full-history dump. In A. Voghera & L. L. Riccia (Eds.), *Spatial planning in the big data revolution* (pp. 16–44). IGI-Global. doi:10.4018/978-1-5225-7927-4.ch002

OpenStreetMapContributors. (2019). *Planet OSM* [Online]. Retrieved from https://planet.openstreetmap.org/

Rousell, A., Noskov, A., & Zipf, A. (2018, June). Data quality concept for e-government web-map based services. In T. Bandrova & M. Konecny (Eds.), *Proceedings 7th international conference on cartography and gis* (Vol. 1, pp. 1314–1604). Sozopol, Bulgaria: Bulgarian Cartographic Association.
Appendix Deployment of an embeddable GSDR instance

The archiving software comprises files (see emgsdr.tar.gz) required for the embeddable services deployment. The proposed services can be easily installed on GNU/Linux and any unix-like systems (e.g., *BSD or MacOS). There are no principal obstacles for installing on Windows systems, but it was not tested. We use the same deployment solutions as we have described in Zipf and Noskov (2018). Further, we provide concrete installation steps for Debian-like systems (e.g., Debian, Ubuntu, Linux Mint, etc.). We use a Debian Jessie system in an ARM cloud.

First of all, we install the standard dependencies provided by a Linux distribution:

$ apt-get install libxml2-dev libgeos++-dev libproj-dev tcllib apache2-dev

Since we need to process multiple concurrent user requests, SQLite database should be configured to use multi-threaded threading mode (Haldar, 2016). Otherwise, there will be errors caused by incorrect processing of multiple requests coming from Apache Rivet. Typically, Linux distributions provide an SQLite library with the "serialize" threading mode by default, which is inappropriate for our system. Thus, we compile SQLite to enable the required multi-threaded threading mode. Because of this, all further dependencies must be built for the source code. The source code is obtained using the following instructions:

$ wget https://www.sqlite.org/2018/sqlite-autoconf-3260000.tar.gz
$ wget http://apache.40b.nl/tcl/rivet/rivet-3.1.0.tar.gz
$ wget http://www.gaia-gis.it/gaia-sins/libspatialite-4.3.0a.tar.gz
$ wget https://www.sqlite.org/contrib/download/extension-functions.c?get=25

Then, all compressed source files need to be extracted using the command
tar --xzvf filename.tar.gz. Next, a user needs to go to every extracted folder and execute the following standard commands:

$ ./configure
$ make
$ sudo make install

WeGovNowConsortium, & OpenStreetMapContributors. (2019). Archived data: Geo-spatial data repository – WeGovNow database. doi:10.5281/zenodo.2454150

Wood, B. (2016). Geojsplit [Online]. Retrieved from https://github.com/woodb/geojsplit

Zipf, A., & Noskov, A. (2018, August). Backend and Frontend Strategies for Deployment of WebGIS Services. In Sixth international conference on remote sensing and geoinformation of the environment (rscy2018) (Vol. 107731C). SPIE. doi:10.1117/12.2322831
For the SQLite library an the procedure is slightly different from others (first, configure the library using the SQLITE_THREADSAFE = 2 mode and, second, go to tea folder to install the required Tcl SQLite library):

```
$ ./configure SQLITE_THREADSAFE = 2
$ make
$ sudo make install
$ cd tea/$ ./configure
$ make
$ sudo make install
```

The SpatiaLite library was configured using this: `./configure --disable freexl`. For ./configure commands, users might apply a `--prefix = /destination/folder` option to install compiled files into a non-standard destination. It is useful to one who does not have root permissions.

Now, a user should install the embeddable system itself. For this, first, she needs to download the archiving software (let us say that it is saved into a gsdrfiles folder) from (Noskov, 2019a), and, second, carry out the installation instructions:

```
$ cd gsdrfiles
$ tar -xvzf emgsdr.tar.gz
$ tar -xvzf nkov.tar.gz
$ tar -xvzf tilescf.tar.gz
$ chmod +x tclshc
$ make NAME=nkov
$ make NAME=tilescf
$ cd/tmp/nkov
$ make install
$ cd/tmp/tilescf
$ make install
```

Next, users need to configure Apache. Supposing that Apache uses the /var/www/html folder for storing web files, 000-default.conf needs to comprise the following virtual host configuration:

```
<VirtualHost *:80>
  DocumentRoot /var/www/html
  LoadModule rivet_module/usr/lib/apache2/modules/mod_rivet.so
  AddType 'application/x-rivet-tcl;charset=utf-8' tcl
  RivetServerConf ChildInitScript "source/var/www/initscript.tcl"
</VirtualHost>
```

The apache2.conf needs to configure Apache Rivet using such code (a user configures concrete values, it depends on available resources):

```
RivetServerConf SeparateVirtualInterps yes
RivetServerConf SeparateChannels yes
<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteCond %{HTTP} !on
  RewriteRule .* https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L,QSA]
</IfModule>

KeepAlive Off
<IfModule prefork.c>
  StartServers 5
  MinSpareServers 5
  MaxSpareServers 10
  MaxClients 150
  MaxRequestsPerChild 1500
</IfModule>
```

Next, databases and server-side code files should be copied to proper destinations:
Finally, a database gsdr_wgn_*.sqlite needs to be copied into the/var/www folder. An instance is ready-to-use after the restarting a web server. Figure 7 demonstrates the usage of the installed system.