

CUSTOMIZED WEBGIS SOLUTIONS FOR EXPOSOMICS

V. Casella^{1*}, M. Franzini¹, M. T. Rocca¹, A. Pogliaghi², N. Fiscante², L. Raso², F. Sapio²

¹ Dept. of Civil Engineering and Architecture, University of Pavia, Italy
(vittorio.casella, marica.franzini, maricateresa.rocca@unipv.it)@unipv.it

² GeneGIS GI Srl, Milano, Italy - (a.pogliaghi, n.fiscante, l.raso, f.sapio)@genegis.net

Commission: ICWG III/IVc

KEY WORDS: exposomics, GIS, time, environment, health, WebGIS

ABSTRACT:

Exposomics is a science aiming at quantifying the effects on human health of all the factors influencing it, but genetic ones. They include environment, food, mobility habits and cultural factors. The percentage of the world's population living in the urban areas is projected to increase in the next decades. Rising industrialization, urbanization and heterogeneity are leading to new challenges for public health and quality of life in the population. The prevalence of conditions such as asthma and cardiovascular diseases is increasing due to a change in lifestyle and air quality. This enlightens the necessity of targeted interventions to increase citizens' quality of life and decrease their health risks. Within the EU H2020 PULSE project, a multi-technological system to assist the population in the prevention and treatment of asthma and type 2 diabetes has been developed. The system created in PULSE features several parts, such as a personal App for the citizens, a set of air quality sensors, a WebGIS and dashboards for the public health operators. Citizens are directly involved in an exchange paradigm in which they send their own data and receive feedbacks and suggestions about their health in return. The WebGIS is a very distinguishing element of the PULSE technology and the paper illustrates its main functionalities focusing on the distinguishing and innovative features developed.

1. INTRODUCTION

Exposomics is a science aiming at quantifying the effects on human health of all the factors influencing it, but genetic ones. They include environment, food, mobility habits and cultural factors. Environmental determinants typically are described by multi-temporal and rapidly varying datasets such as air quality maps. Health data are inherently dynamic, as well. Healthcare agencies typically output prevalence maps every year or more frequently.

Datawarehouse for exposomics applications must therefore be able to store time-dependent, rapidly varying and very heterogeneous datasets (Bettencourt, 2014). They normally are big data repositories.

The percentage of the world's population living in the urban areas is projected to increase in the next decades. Rising industrialization, urbanization and heterogeneity are leading to new challenges for public health and quality of life in the population. The prevalence of conditions such as asthma and cardiovascular diseases is increasing due to a change in lifestyle and air quality. This enlightens the necessity of targeted interventions to increase citizens' quality of life and decrease their health risks. To this end, the European Commission project named PULSE (Participatory Urban Living for Sustainable Environments) unifies several institutions around the world and works with the municipalities of 7 cities – Barcelona, Birmingham, Paris, Pavia, New York, Singapore and Keelung – to develop a multi-technological system to assist the population in the prevention and treatment of asthma and type 2 diabetes. The system created in PULSE features several parts, such as a personal App for the citizens, a set of air quality sensors, a WebGIS and dashboards for the public health operators. Citizens are directly involved in an exchange paradigm in which they send their own data and receive feedbacks and suggestions about their health in return.

The WebGIS is a very distinguishing element of the PULSE technology. It is based on the open source technology (Java and

Tomcat for backend, JS and OpenLayers for the frontend, Geoserver, PostgreSQL/PostGIS DBMS) and includes highly innovative features. First of all, each data item has a time tag and the system is able to easily navigate in time, switching on and off the appropriate layers. Another distinctive feature of the Pulse WebGIS is related to its data management and data configuration capabilities, that make the platform very flexible and generic. In fact, it not only provides visualization access to geospatial data, but it also offers the necessary features to manage the data lifecycle and the authoring tools to administer layers, create or modify existing maps and change their configuration.

In particular, a specific data loading management component was designed to allow the user to add new layers into the system in a quick and straightforward way, simplifying and automating several steps that normally the user need to perform manually, such as data validation, conversion/loading into the database and publication to the GIS server.

A powerful configuration tool, meant to be used also by non-professionals, allows to organize the layers in maps and to create/customize a WebGIS: in particular it is possible to configure the data in the menu (Table of content), dynamically load new layers in the Viewer and also define the various widgets for the map (e.g. temporal navigation, attribute table, charts, search tool, printing, double map visualization, etc). Security aspects can also be easily managed, and access / visibility to the resources can be granted or restricted, making it possible to either publish public or protected content according to the specific need. The paper illustrates its main functionalities focusing on the distinguishing and innovative features developed.

2. PULSE WEBGIS CONCEPT

The architecture of PULSE WebGIS has been designed and the system has been implemented with the following key concepts in mind:

- Scalability: the system is designed to be able to manage a variable number of layers and also a variable number of

users. The architecture and the technology that was chosen allow to both scale the system vertically and horizontally. If needed, the components of the system can scale out, and additional instances can be added in order to comply with an increased volume of requests or data. All the components can be clustered and load balanced, and scalability can be achieved at both the database level, at the application server level and at the cartographic server level.

- **Modularity:** the system is divided in modules and it can be progressively enriched and extended with additional functionalities and tools without having to modify the base architecture. The concept of modularity was followed on both the design of the architecture (each part of the system is dedicated to a specific task and independent from the others), and in the physical implementation. From a software engineering point of view, the system is organized in a set of independent functional modules, each focusing on a specific task. This structure allows also to replace and/or improve parts of the system with a minimal impact on the others, which remain mainly unaffected.
- **Interoperability:** the system is designed to be as much interoperable with other external systems as possible. Technologies, data types and protocols follow the best practices and established standards. Components are loosely coupled and the communication between them is based on open, non-proprietary, interoperable protocols. They don't depend on any vendor-specific libraries or functionalities and are almost fully platform-independent. GIS data are exposed to the users according to international geographical standards (OGC), while the communication between the components is based on web-services and JSON-based messages.

From a logical point of view, the WebGIS architecture (Figure 1) is divided in: WebGIS viewer, GIS backend, PULSE WebGIS Database and Cartographic Server.

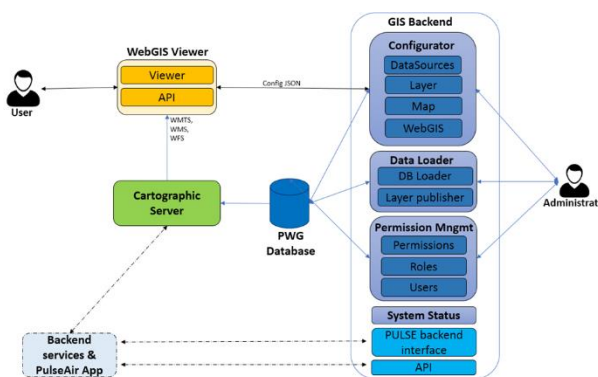


Figure 1. The logical architecture

The Viewer represents the WebGIS navigator. It is the main interface for visualizing and analysing the geographical data of the different pilot cities. The GIS backend application is the administrative component of PULSE WebGIS, that handles most of the management logic. The PWG database is used for storing data. It is based on a Spatial Database Management System (DBMS), that is a database optimized for storing and querying data that represents objects defined in a geometric space. While typical DBMS work with various numeric and character types of data, Spatial Databases present additional functionality in order to process data related to objects in space, such as points, lines and polygons. Most of the spatial databases offer functions to perform spatial operations such as coordinate system conversions, buffer, geospatial analysis, distance calculations and so on. They also provide ad-hoc data structures, primitives

and operators specifically designed to deal with spatial data in a very efficient way (e.g. spatial indexes, geometry types, bounding-box comparison operators, etc.). The Cartographic Server consists of a server application that is specialized in publishing geospatial data coming from heterogenous data sources and to make it available through open protocols via standard cartographic web services. In PULSE WebGIS, the role of the Cartographic Server is to fetch vector and raster data from the storage layer (DBMS or filesystem) and provide it, in a suitable format, to the Viewer for query and visualization.

3. WEBGIS VIEWER

The WebGIS Viewer is one of the main components of PULSE WebGIS. It is used to access and visualize the cartographic data loaded and published into the system. It is a Web-based application and can be accessed using a browser.

The Viewer has all the characteristics and functionalities of the current state-of-the art WebGIS applications, plus some innovative features.

3.1 Viewer common functionalities

The dashboard has all the main and common functionalities such as table of content (TOC), coordinate system toolbar or base map selector.

TOC is an organized list of the layers currently loaded on the map. It follows a hierarchical structure: layers are organized in groups and it is possible to both change the order of layers within the group and change the order of the groups themselves. In this way it is possible to control the visualization of the layers in the map and consequently define which layer sit on top of which other. Furthermore, coordinate system toolbar shows the map coordinate reference system, the current coordinates of the mouse pointer and the current scale (zoom level). Finally, the base map selector allows to change the current background using some WMS or WMTS layer such as OpenStreetMap or Bing. Figure 2 shows the WebGIS layout where it is possible to see some of these widgets.

However, there are some important and innovative functionalities that will be detailed in the next paragraphs.

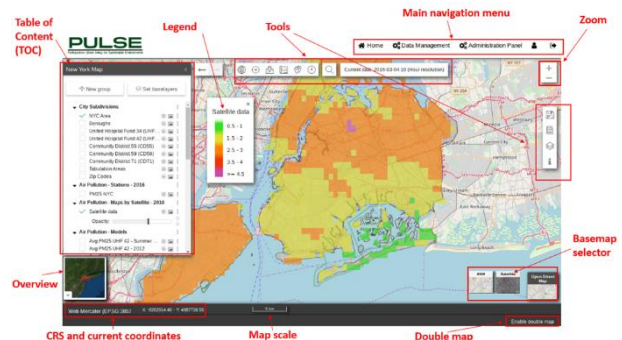


Figure 2. The WebGIS Layout

3.2 Double maps visualization

One of the distinctive features of the Pulse WebGIS is the possibility to display two maps side-by-side. To implement this functionality, the structure of the viewer has been designed according to a hierarchical model, that allows a composable approach. The WebGIS can contain one or two maps; each single map is configured independently from the other and each map defines its own layers and widgets: this allows independent configuration and the possibility to show different data. These

maps are grouped together in a single WebGIS, that allows the user to simultaneously display both. It allows the users to quickly visually compare the status of different phenomena in the same city to find relations, disparity or patterns. Moreover, when coupled with the Time Manager, it allows not only to show two different phenomena at the same time, but also to show the same phenomenon at two different times. This capability is particularly interesting in exposomics where different data, environmental and of public health for instance, must be compared.

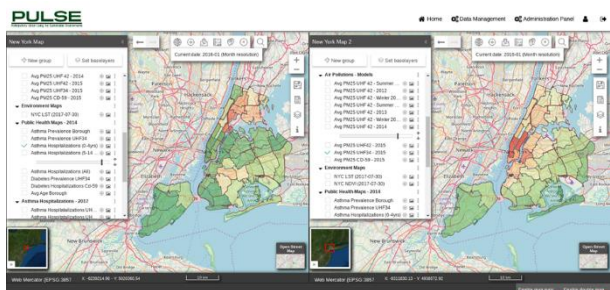


Figure 3. Double map, synched view

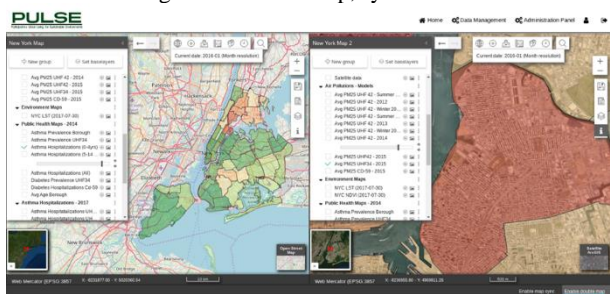


Figure 4. Double Map, independent view

Figure 3 shows one of these examples, where a map showing the average air quality for a defined time is visually compared with a map showing the asthma hospitalizations in the same period. It is possible to have both a synchronized view between the two maps or an “independent view” (Figure 4). With the “synchronized view”, the area shown on the two maps is kept in-synch as the user interacts with the maps: zooming and panning a map will automatically cause the other map to follow the same movements and the two maps will always point to the same position in space with the same zoom level. If the synch-view is disabled, the user can navigate the two maps in a complete independent way.

3.3 Single feature query and attribute table widget

PULSE WebGIS allows the user to query a layer and obtain the information of the elements of the layer situated at the selected coordinates. In addition to this simple query functionality, that is traditional and commonly found in most WebGIS, PULSE WebGIS also supports the ability to display the Attribute Table of vector layers (i.e. layers having alphanumeric information associated to the feature geometries) and simultaneously show the tabular and geographic components of a layer, in a way similar to what is employed by desktop GIS systems. Tabular data is shown in a separate widget. Since the amount of alphanumeric data related to a layer may potentially be very large (e.g. hundreds or thousands of rows, each consisting of dozens of columns), data retrieval and visualization is done using server-side pagination, so that processing is done on the server and the strain on the client is minimized. When a feature (i.e. a row) is selected on the table, the map automatically zooms to the corresponding geometry and the relevant feature is highlighted.

Public health data are particularly rich, and this functionality allows to easily navigate into them thanks to a readable interface. Figure 5 shows, as example, the feature query for the New York City asthmas data: in the bottom part, the classical attribute table is shown, whereas on the right one the implemented widget.

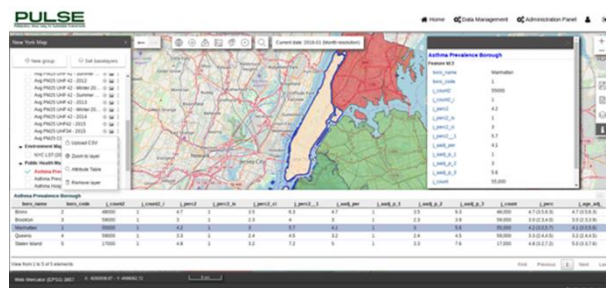


Figure 5. Query feature and attribute table

3.4 Attribute table charts

In addition to the plain traditional tabular format, PULSE WebGIS allows to display charts and diagrams. Sometimes this is helpful, as it makes it easier to interpret the data of the table of attributes. Within it, it is possible to define the charts to be shown for each layer in the system (type of chart, x and y-axis, data, title, etc). If the attribute table of the layer being shown has some charts defined, it is possible to display them in a chart widget, as illustrated in Figure 6, where, as an example, the asthma prevalence per district distribution is shown in a histogram.



Figure 6. Attribute table charts

3.5 Temporal navigation

In addition to normal layers, PULSE WebGIS is also able to show time-series of layers (so-called “temporal layers”), allowing to navigate data in time and display in the map only the data and attributes corresponding to the specific moment being selected. The implemented time manager widget is the tool that allows to perform such an operation.

With the time manager, the user can specify a date and choose a temporal resolution (e.g. hour, day, month). Once the interval is selected, then all the “temporal” layers defined in the map are refreshed according to the specified interval. The non-temporal, normal layers remain unchanged in the map. Time manager widget is shown in Figure 7: in the top panel, the current interval and temporal resolution is displayed. The widget itself (in the bottom) allows to navigate in time, selecting the intervals where the data of the temporal layer are available. In the example, the user is navigating the raster time series of “Satellite Data” layer in New York City and is currently displaying the data referring to April 4th, 2016 at 10AM.

Time-series layers are modelled by having a single “logical” layer (that is displayed in the TOC), that is associated to a list of

n physical layers. The logical layer just acts as a container for the physical layers providing a handle to the data, while the physical layers are those effectively containing the data. The selection of a specific physical layer depends on the interval/resolution chosen by the user (Figure 8).

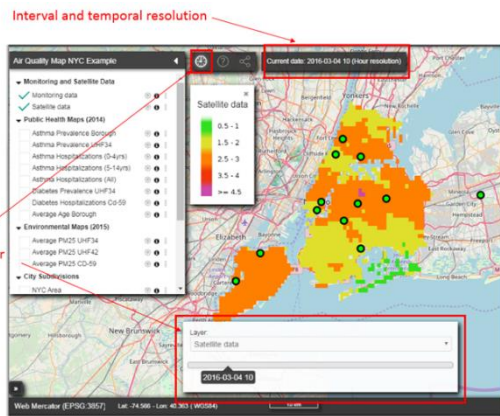


Figure 7. Time Manager Widget

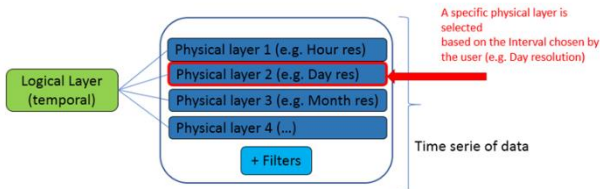


Figure 8. Temporal layers modelled as logical and physical layers association

Due to the differences between the raster and vector model, temporal navigation internally works differently, and the two types of data are managed with two approaches, as illustrated in the following Figure 9.

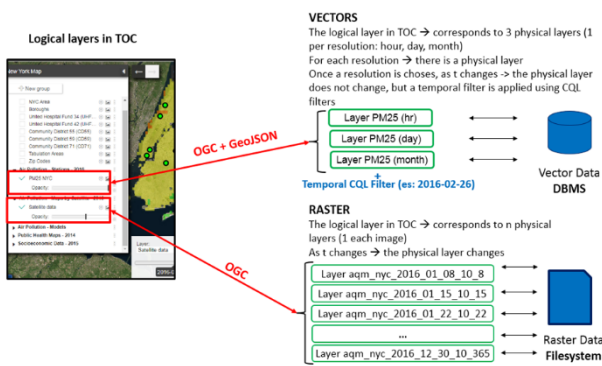


Figure 9. Temporal layers mapping to physical layers

In the case of vector time-series, the logical layer in the TOC (for instance, let's suppose it is PM25 concentration in New York City) corresponds to 3 physical layers, each of which provides the data according to a specific temporal resolution (i.e. hour aggregation, day aggregation, etc.). The correct physical layer the system uses to retrieve the data is chosen according to the temporal resolution that the user has selected: for instance, if the user wants to look at the daily averages of PM25 concentrations in New York City, the "day aggregation", the "day resolution" physical layer underneath the "PM25" logical layer is used. Once the user has chosen the temporal resolution, as he changes the interval (e.g. selects a

specific "day" if he is using the "day resolution"), temporal filter is applied to the physical layer, in order to obtain only the data belonging to the requested interval.

Physical temporal layers can either be WMS layers provided by an OGC-compliant WMS Cartographic server, or can be GeoJSON layers, provided by a general Web-Service. In case a WMS is used, the time filter is passed to the server by using standard OGC CQL (Common Query Language) syntax, that makes the Viewer independent from the specific WMS server implementation.

In case of GeoJSON data (such as, for instance, the data retrieved from the Pulse Backend Interface), a single WebService endpoint is used, and the temporal filter condition is expressed using simple Http request parameters.

In case of raster time-series, the logical layer in the TOC corresponds to n physical layer on the system, where each of them represents the actual data. In this case, the temporal resolution is fixed and depends on the data (for instance for NYC Air Quality Satellite images, there is 1 image per week and each image refers to average measurements taken in 1-hour period). When the user changes the time interval, the raster corresponding to that specific interval is chosen and retrieved. For raster time-series, WMS layers are used.

Examples of vector time-series and raster time-series are shown in Figure 10.

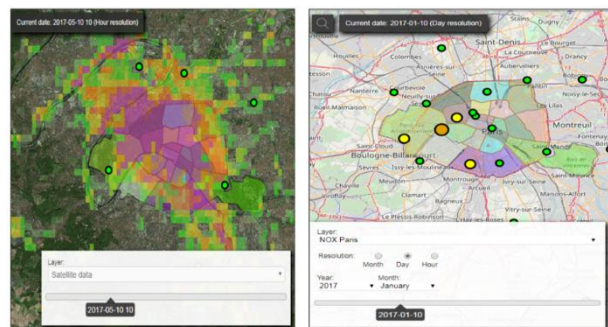


Figure 10. Time-series data examples: Air quality Satellite (raster, left) and Air Quality from Monitoring stations (vector, right)

3.6 Client-side external CSV data loading

As previously described, the WebGIS Viewer allows the user to display the attribute table of vector layers, showing the attributes defined in the layer schema and, for each feature, the corresponding values. Moreover, a feature of the viewer allows the user to dynamically load new data in the WebGIS from a CSV (comma-separated value) file, using the geometry of another layer.

This operation is performed on-the-fly in the client-side and the resulting layer is treated as if it were loaded from the server, meaning that the other existing functionalities dealing with attributes are supported as well (e.g. charts, query info, etc).

An example is provided here to better clarify the functioning of this feature. Suppose that a user, on his computer, has a CSV file containing the value of some parameters (e.g. asthma hospitalizations for 2019, but any other parameter will do) aggregated by city district.

With this functionality, the user can load the CSV and associate it with the geometry of the District layer, thus creating a new layer, whose geometries are districts, and whose attributes are those expressed in the CSV file.

The following Figure 11 shows the Upload CSV dialog. Note that it is necessary to select which are the fields to use in both the existing layer (e.g. the District code for the District layer) and the

CSV file (e.g. the “code” attribute) in order for the widget to be correctly able to join together the two data. Once the new layer has been added into the viewer, it is possible to change its style using the thematization widget, explained in the next section.

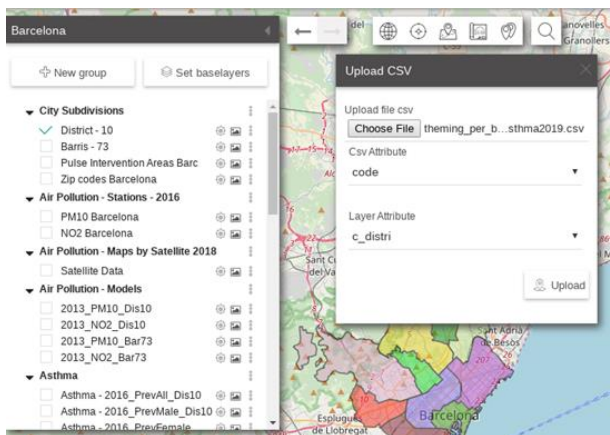


Figure 11. Upload CSV dialog

3.7 Thematization Widget

This component allows the user to customize the style of the dynamically loaded layer, such as the CSV external ones (Section 3.6). The user can select the attribute to use for creating the style (e.g. “Asthma prevalence”) and then define the styling rules.

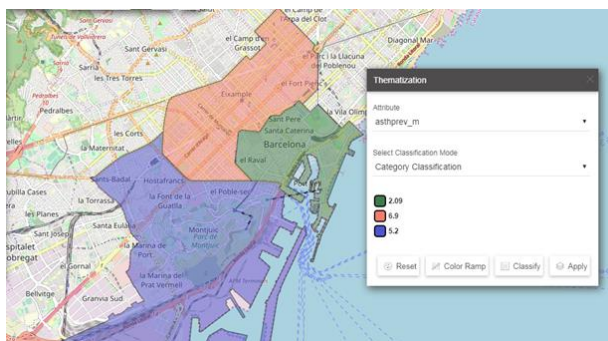


Figure 12. Thematization widget (by category)

The widget currently supports two automatic classification modes: category and classes. These modes are commonly found in GIS desktop clients. The category classification generates a styling rule for each unique value of the selected attribute, and it is normally used for textual values. The classes classification divides the data in the specified number of classes. The division can be done either by dividing the attribute space in sets having the same interval lengths, or by quantiles. In the first case, intervals have the same length but may contain different number of features, while in the second case the features are split evenly among the classes, therefore classes may have different interval lengths. Category thematization is shown in Figure 12, while classes thematization is shown in Figure 13.

The thematization widget allows to create a set of “Styling rules” that can be applied to the layer. The mechanism that has been developed in order to dynamically create and apply styles to WMS layers is described in the following paragraph.

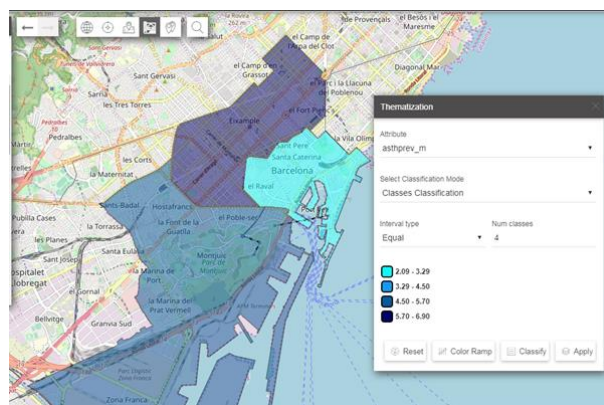


Figure 13. Thematization widget (by classes)

3.8 WMS Layer Dynamic Thematization

Since a layer served by a cartographic server via WMS protocol is constituted by images, the styling information on how the data needs to be styled is normally defined in the cartographic server. To be able to allow the user to dynamically define a new style for a layer and apply it on-the-fly, the following mechanism has been designed and developed within the PULSE WebGIS. This mechanism allows to define a new style and pass it to the server so that it can use it instead of using the associated one.

The process is illustrated in Figure 14 and the steps are briefly described here:

1. rules on the style are coded in a JSON-based data structure on the client-side and passed to a webservice in the backend;
2. the webservice parses the JSON structure and creates a standard OGC SLD style document associated to the styling request. A token is generated, associated to the SLD and added to a cache;
3. the token corresponding to the generated SLD is returned to the client;
4. the client performs the usual WMS requests to the cartographic server, setting the SLD parameter to the URL of a webservice of the WebGIS backend that provides the SLD document. This URL contains the token as a parameter;
5. the cartographic server receives the WMS request, parses the URL contained in the SLD parameter and requests the style from the WebGIS backend using the token;
6. the WebGIS backend returns the SLD associated to the token;
7. the cartographic server applied the SLD to the layer, generates the WMS image and streams it to the client for visualization.

The caching mechanism allows to avoid generating a new SLD document each time the client performs a WMS request, speeding up the whole process.

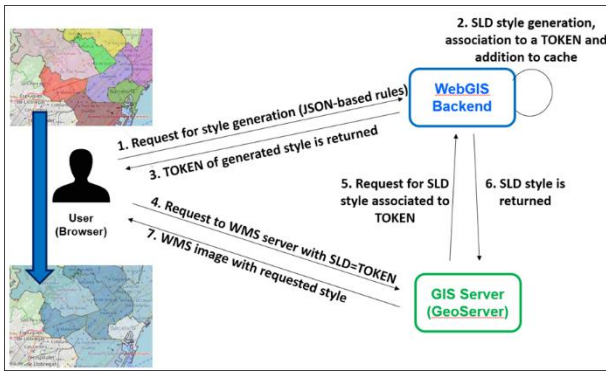


Figure 14. WMS Dynamic Thematisation

3.9 Interface to Monitoring Station data

The WebGIS allows to display the information coming from the different monitoring stations deployed in the pilot cities. Indeed, one of the main project technological item is the implementation of dense network of sensors for air quality. Monitored data are obtained by the backend services using different protocols according to the sensor type (e.g. Dunavnet, PurpleAir). So far, more than 50 million measurements have been collected. The WebGIS allows the user:

- to see the measurements for a specific parameter related to a moment in time or the most recent ones (Figure 15);
- to query a specific monitoring station and consult the values of all the parameters it measured;
- to see charts related to the trend of a specific parameter, with the possibility to filter the time interval and aggregate the values according to a time resolution (i.e. raw data, data aggregated by hour, day, etc);
- to see diagrams related to 2 different parameters in the same time, making it easier to visually compare their trends (Figure 16).

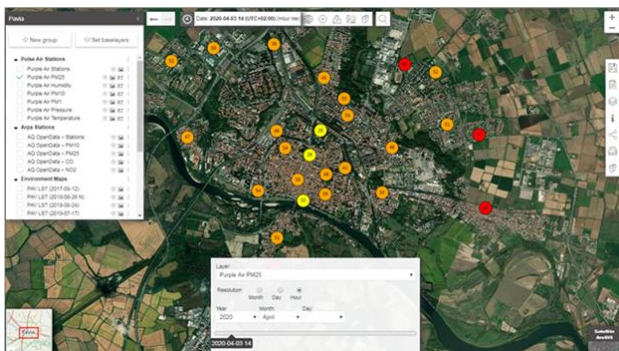


Figure 15. PurpleAir PM 2.5 measurements in Pavia



Figure 16. Temperature and PM25 trends for a PurpleAir monitoring station in Pavia

3.10 WebGIS Configurator and Data loading Module

The WebGIS not only includes a component (i.e. the Viewer), that provides visualization access to the geospatial data, but it also includes a Configurator component, meant to be used also by non-professionals, that handles all the management aspects of the platform and provides a set of tools to manage and administer the datasets.

Specifically, it allows to:

- create new WebGIS(es);
- customize maps, layers and widgets;
- manage the life-cycle of geographical data (e.g. loading new data, publishing on the cartographic server, registering external data already published on external data sources, etc.)
- perform general administration (permissions, roles, users, logs, etc.)

It is composed of different modules (some of which available to system administrator only), that allow to create and/or configure, in a highly customizable way, the various parts of the WebGIS: data, maps, layers and data-sources.

In particular, it is possible to customize the WebGIS of a pilot city, by removing existing layer from the table of content, or selecting the new ones to add from the layer catalog list (i.e. the list containing all the layers loaded in the system). It is also possible to define the layer organization in groups and relative ordering within the table of content.

Figure 17 shows the layer catalog containing the list of all the layers loaded in the system: a layer can be added to the TOC by choosing it. Once done, the current map can be saved.

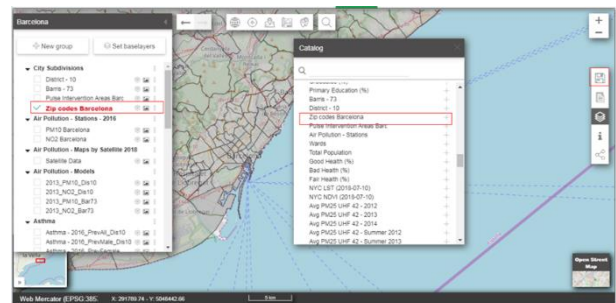


Figure 17. A layer is added to the existing map

The Configurator and the Viewer are meant to be loosely-coupled and their interaction is based on an exchange of JSON-based messages: the Viewer receives a message containing all the information required to correctly load and display the specific WebGIS of the pilot city that was requested (e.g. maps, layers, widgets, temporal configurations, charts, etc.), as exemplified in Figure 18.

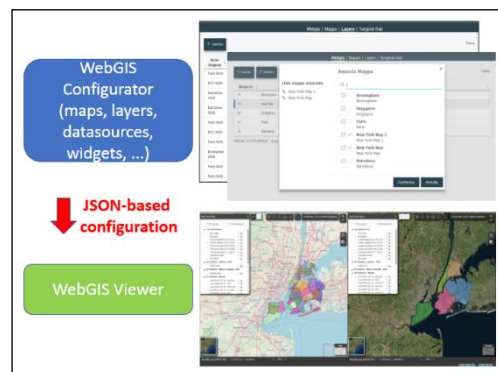


Figure 18. Viewer and Configurator interaction

A very distinctive feature of the Configurator is related to the Data Loading module, that allows the administrator to add new vector data into the system in a quick and simplified way, automating some steps that otherwise the user would have to perform manually.

These steps normally involve:

- performing consistency check to ensure syntactic coherency of the data (such as allowed coordinate reference system, datatypes, value encodings, etc.)
- converting the shapefile into a table and loading it into a spatial database
- publishing the layer into a Cartographic Server
- associating the layer to a map so that it can be displayed in the Viewer

They also often require the use of different tools (such as a Desktop GIS to load the data in the database, or accessing the Cartographic Server User Interface to manage the publishing) that further complicate the process and make it less straightforward.

The Data Loading Module provides an automated workflow to handle these operations: the user can load and manage a layer from a simple custom-made graphical interface; complexities and technicalities are hidden and the actions are executed automatically.

The module has been designed with a flexible structure, in a way that it could be adapted to other contexts, but the actual operations (loading and publishing) are specific for the technologies employed in the project (i.e. PostGIS and GeoServer). The current supported format for automatic loading is the shapefile, since it is a very popular geospatial vector format.

The following Figure 19 graphically illustrates the workflow.

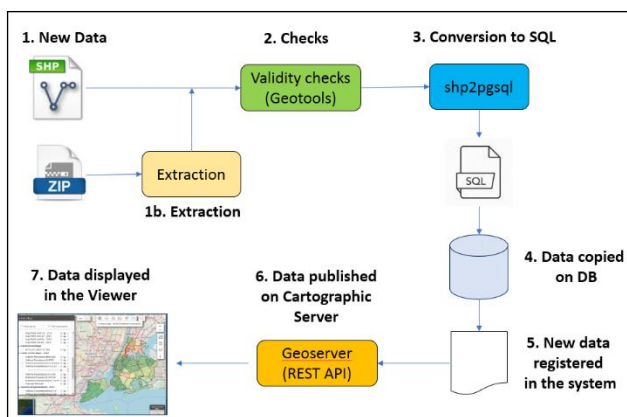


Figure 19. Data Loading Workflow

Several tools are involved in the process: the Geotools Java library is used to perform validity checks on the data; shp2pgsql is used to convert the shapefile into a PostGIS-compliant SQL, while the Geoserver REST APIs are used for the publication. A Java library has been developed specifically for the interaction with Geoserver, providing a set of easy-to-use methods that allow to perform operations on it without having to deal with low-level communication issues.

4. DISCUSSION

The development of the Pulse WebGIS followed an iterative approach: different versions of the system have been developed and functionalities/data have been added at each cycle.

At the beginning of the project, the first versions were implemented and deployed on a development environment; later,

the platform was installed and configured also on the cloud Teralab infrastructure, one of the partners of the project.

Initially, the efforts concentrated on the WebGIS Viewer and the first prototype versions included only some data for New York City and provided basic functionalities such as zoom, pan and layer visualization. Subsequently, a specific WebGIS was created and configured for each pilot and their relevant data were loaded. More advanced functionalities were developed and added to the viewer (time manager, side-by-side visualization, feature query, attribute table, etc.), and the Configurator module implemented. Historical data and the interface to the monitoring station was also added.

Since 7 pilot cities are involved in PULSE, the WebGIS has been designed to be as flexible, as general and as data-agnostic as possible, in order to support an arbitrary number of cities, each with its own specific data sets. At the database level, pilot data were separated in different schemas, in order to maintain the system more manageable. The Configurator module was extensively used during data configuration, and it proved to be quite powerful, simplifying the addition and configuration of new layers. The Pulse WebGIS has been designed to be used both as a standalone application and also as an integrable component that can be included or embedded in external applications.

The main role of WebGIS in the Pulse architecture is to provide access and visualization to geo-data, but it also provides geospatial services to the other components of the system.

To be able to support the various request types from heterogeneous applications, different communication interfaces and protocols have been implemented. The WebGIS supports the standard OGC protocols, and also uses various custom Web-services to communicate with the Pulse components. Moreover, a Javascript API was designed and implemented to be used when the Viewer needs to be embedded in an external Web Application; the API allows this external application to interact directly with some of the functionalities of the Viewer.

The API has been designed to be used primarily by the PULSE Dashboard, but, being generic, it allows the WebGIS Viewer to be integrated in any other external application.

5. CONCLUSION AND FURTHER ACTIVITIES

Within the H2020 PULSE project, an innovative WebGIS has been implemented. It is based on open source software technologies and is characterized by a number of advanced features:

- side by side synchronized visualization;
- fully fledged time manager;
- capability to receive and integrate real time data streams;
- advanced user management;
- capability to load new tabular datasets and spatially enable them.

There is another feature under development: full integration of personal exposure data. Within the EU H2020 PULSE project, an innovative mechanism for the individual and dynamic assessment of exposure to air pollution has been implemented. It is performed by using data coming from a on-purpose deployed dense network of low-cost air quality monitors and a special App which is capable to track individuals' paths. Results will be available through the WebGIS at two levels: aggregated for general users and detailed for policy makers.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the municipality of Pavia, ARPA Lombardy and all the PULSE consortium for their

cooperation and contributions to this study. This work has been funded by the European Commission Horizon 2020 Framework programme under grant agreement GA-727816.

REFERENCES

Anandan, C., Nurmatov, U., Van Schayck, O.C.P., Sheikh, A., 2010. Is the prevalence of asthma declining? Systematic review of epidemiological studies. *Allergy*, 65(2), 152-167. doi.org/10.1111/j.1398-9995.2009.02244.x

Bettencourt, L.M., 2014. The uses of big data in cities. *Big data*, 2(1), pp.12-22.

Guarnieri, M., Balmes, J.R., 2014. Outdoor air pollution and asthma. *The Lancet*, 383(9928), 1581-1592. doi.org/10.1016/S0140-6736(14)60617-6