

WATER DISTRIBUTION MODELLING IN THE CLOUD: WEB-GIS IMPLEMENTATION

Gareth LEWIS¹, Bret SNIDER¹, Lydia VAMVAKERIDOU-LYROUDIA²,
Albert S. CHEN¹, Slobodan DJORDJEVIĆ¹, Dragan A. SAVIĆ²

¹ Centre for Water Systems, University of Exeter. Harrison Building, North Park Road, Exeter

² KWR Water Research Institute. Groningenhaven 7, the Netherlands

ABSTRACT

Hydroinformatics tools are commonly used for modelling and management of water systems. Those activities involve using spatially and temporally varied information collected from various sources and often available in different data formats. Creating these tools is often the domain of specialist software developers and their effective use by water practitioners is associated with a steep learning curve. Software or applications such as open-source Geographic Information Systems (e.g., QGIS) and water distribution system modelling software (e.g., EPANET) have been developed to support the analyses and decision making. However, these solutions have some critical weaknesses: 1) it can take users considerable time to become proficient in using a given application, 2) applications generally assume a level of domain-specific knowledge, 3) applications may require customisation through plug-ins to provide suitable information and 4) applications are often tied to specific hardware / operating system configurations. To address these issues, the EU-funded aqua3s research project had a goal of developing a cloud-based platform integrating QGIS and EPANET visualisation and modelling capabilities, using the cloud/web to deliver functionality.

This hydroinformatics approach resulted in methods capable of integrating and visualising information to

support water management, specifically developing a web-based EPANET simulation and visualisation for large water networks (c30,000 network nodes and links). The resulting application demonstrates how data can be processed and balanced between client and server to minimise client loading and maximise responsiveness. The process of creating the application and the tools used in this project are also described in detail to allow reproducibility of the research work.

Keywords: Hydroinformatics, open source, Web-GIS, EPANET, GIS, Shapefiles, Visualisation

1 INTRODUCTION

Hydroinformatics, as a scientific study of this complex system takes a deliberately interdisciplinary, sociotechnical approach (Makropoulos and Savic, 2019), bringing together water science, data science and computer science. Visualisation of spatial water data is an important way of presenting and communicating data among water professionals and with various stakeholders. Computer-based systems for capturing, storing, checking, and visualising spatial data, Geographic Information Systems (GIS), are often used to present and communicate water-related data (Maidment, 2002). Urban water distribution systems (UWDS), which are highly complex, large-scale, non-linear, time-varying, socio-techno-economic networks, are part of the essential urban infrastructure (Savic and Banyard, 2011). Due to their importance for public health, once a UWDS is in operation various investigations can be done either directly on the physical system or by using models. EPANET is a widely used open-source computer-based simulation software developed by the US Environmental Protection Agency (Rossman, 2000). Using network data, demand

Podaci o prihvatanju članka

Primljen: 4.5.2023.

Ispravljen: -

Prihvaćen: 14.6.2023.

Kontakt: d.savic@exeter.ac.uk

data and operational data, EPANET software produces results related to the performance of the network, e.g., water demand met by the system, pressure deficiencies, leakage levels, etc.

While EPANET provides a basic Graphical User Interface (GUI), it does not have GIS capabilities, which limits spatial visualisation of the UWDS data and model results. Attempts have been made to link GIS with the EPANET hydraulic engine, but most of those have been delivered through a single desktop application (Bartolín et al., 2008; Ramesh et al., 2012; Ahmadullah and Dongshik, 2016). Recent work by Bayer et al. (2021) demonstrated an EPANET model repository that used open-source building blocks to create an open-web application for sharing EPANET models. Due to that primary aim, the application has limited map manipulation and visualisation capabilities.

QGIS (2023), a free and open-source GIS system, and EPANET, a free and open-source UWDS modelling software, are often the choice of professionals interested in visualisation and simulation computer platforms. QGIS allows users to visualise complex environments using a collection of layers built on top of vector or coordinate system (XYZ) tile sets to view data graphically with map and/or satellite image contexts. Likewise, the EPANET application enables users to interactively build and simulate water networks using the built-in hydraulic solver to provide accurate simulation results.

Whilst QGIS and EPANET can be incredibly useful, they have steep learning curves and, in the case of EPANET, only provide limited platform (Windows) support. For developers of water management systems, it is desirable to take core functionality from QGIS (layered map and data visualisation) and EPANET (water network visualisation and simulation) and present it to users in a way that can: 1) be used without the need for bespoke computers and 2) be used with little training or support.

The “aqua3S” project (funded by the European Union’s Horizon 2020 research and innovation program, 2019-2022) focused on creating strategies and methods enabling water utilities to easily integrate solutions regarding water safety through the combination of novel technologies in water safety and the standardisation of existing sensor technologies (aqua3S, 2023). One of the goals of aqua3S was to develop a GIS visualisation application that runs UWDS modelling analyses and presents them within a web browser context. This is to

be done via the client-server architecture, which breaks down tasks between clients (e.g., a personal desk computer) and servers (hosted centrally) that are linked by a computer network (e.g., local area network or over the Internet). The resulting software is aimed at water management practitioners who do not have the software skills to develop such applications but could benefit from the availability of such an integrated application that can work on any basic computer that can connect to the Internet regardless of its processing capabilities.

The remainder of this paper is organized as follows: Section 2 describes the characteristics of the web-based tile-map visualisation systems and their suitability for the development of a platform for EPANET modelling and results visualisation; Section 3 describes the process of converting spatial data from shapefiles to data formats available for web-based tile-map visualisation; Section 4 demonstrates how EPANET simulation runs are visualised, and Section 5 provides conclusions with some future research challenges.

2 WEB-BASED TILE-MAP VISUALISATION

The best-known mapping services for the web, such as Google Maps, Microsoft Bing and Apple Maps, use maps to display spatial information in a browser. These web-based tile maps split geographic maps into collections of tiles allowing users to scroll over a map giving the impression of a seamless map surface. Map tiles are subdivided into increasing levels of detail, to create the effect of zooming into or out of the map (Figure 1).

Software developers have various solutions available for the development of tile-map visualisation tools. These tend to deliver similar functionalities but with different access policies for each of the solutions. Our initial investigation focused on frameworks provided by Google Maps (2023), Mapbox (2023), leaflet.js (2023) and MapLibre (2023a). However, due to restrictive terms of usage, the Google Maps platform was not tested further.

The remaining three frameworks were similar in their conceptualisations, which is not surprising given that MapLibre is an open-source drop-in for Mapbox and Leaflet.js also appears to have had some involvement with Mapbox. The frameworks work on the concept of a view being a collection of layers, which are either provided by the map tile server (ground, buildings, roads, labels, etc.) or provided by users, with five distinct components - default marker, GeoJSON

geometries, animated images, video and WMS, which form the basis for user-defined visualisations.

The default 'marker' visualisation type can be thought of as the de facto visualisation/interaction style for web-

based tile maps and consists of a 'pin' in the map at a given location (Figure 2). Typically, the pin can be clicked by a user to reveal location-specific information.

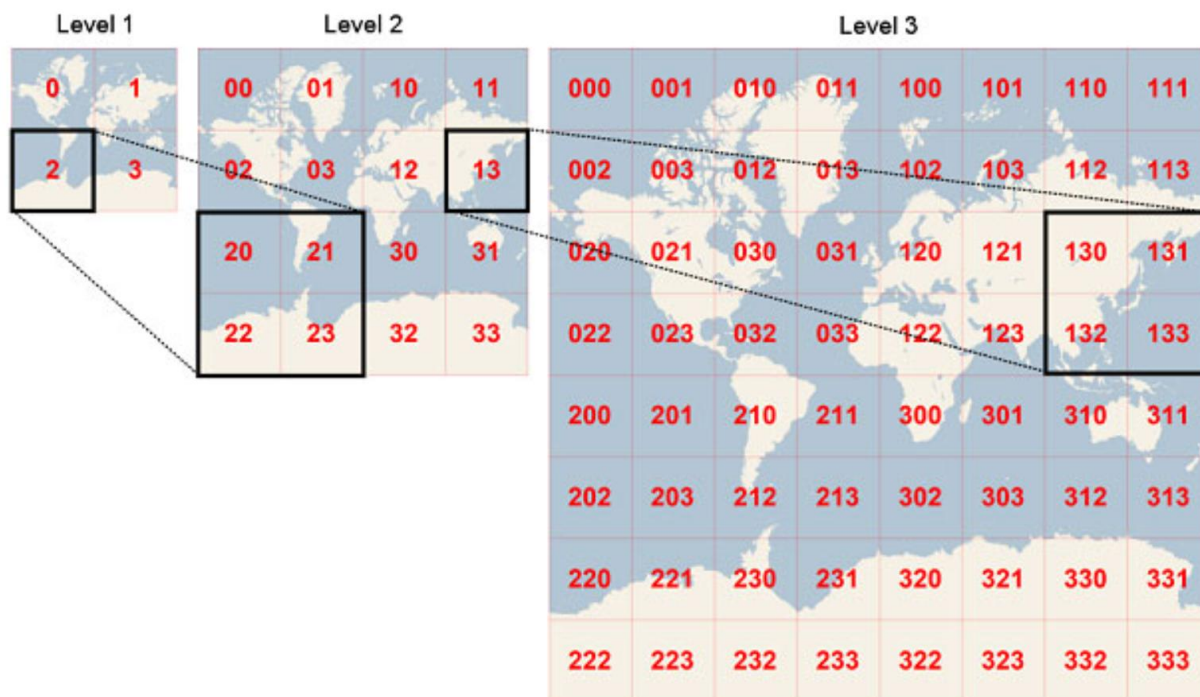


Figure 1. Tile maps showing levels of zooming Bing maps tile system (2023)

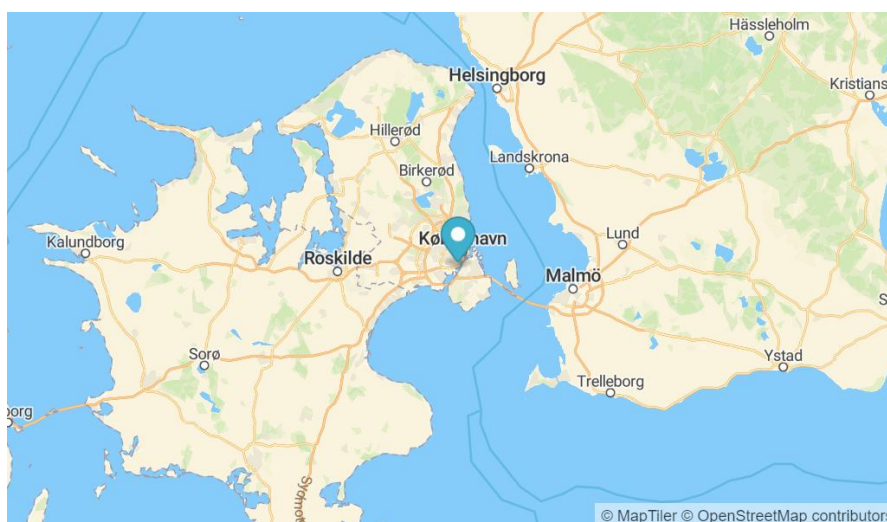


Figure 2. An example of default marker ('pin') visualisation (MapLibre, 2023b)

The GeoJSON geometry visualisation type is a more open-ended approach to visualisation/ interaction and can be considered as a data transport into the tile-map solution. GeoJSON (2023) supports a range of geometry types (points, lines and polygons) allowing users to create pin-like interactions with point and line types and/or region-based interactions with the polygon (Figure 3).

The animated image visualisation type can be used in ‘weather map’ style visualisations, where animated data is overlaid onto a map. Care should be taken that the pixel size of animated data may not work well with different map zooms, the MapLibre (2023d) example is a good case in point (Figure 4). The image of 640 x 456 pixels is projected onto an area of about 600 km x 600 km, giving a resolution of one pixel per km, which results in a good presentation from distance, but detail is lost when the user zooms into the map.

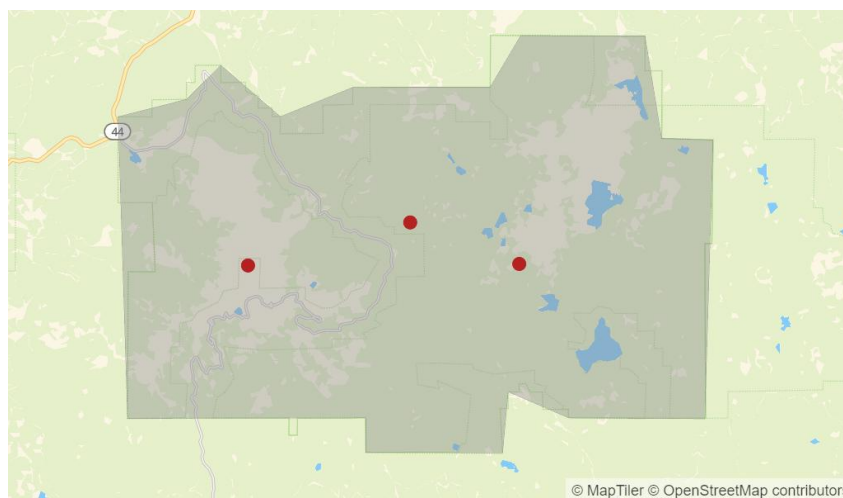


Figure 3. An example of GeoJSON geometries visualisation (MapLibre, 2023c)

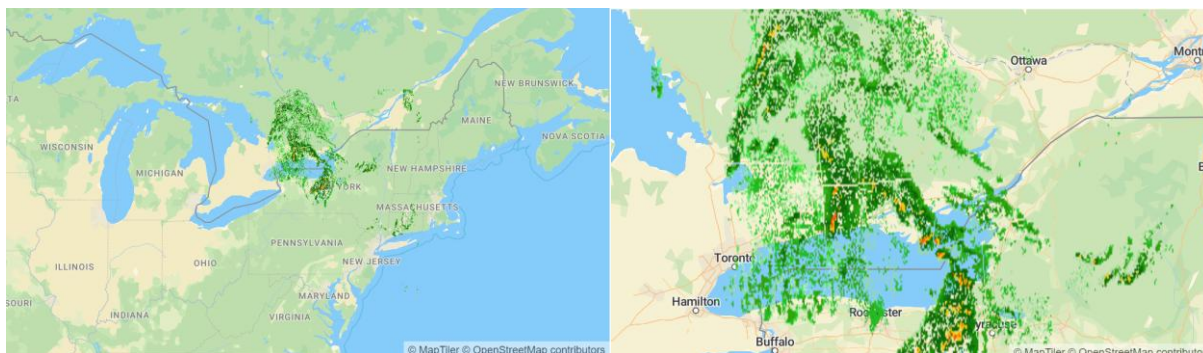


Figure 4. An example of animated images: (a) a view from distance, and (b) a zoomed-in view (MapLibre, 2023d).

The video visualisation type provides a similar solution to animated images, but the use of a video format rather than a collection of images allows for more frames of animation for a given data footprint, which may be useful for managing collections of files for fluid animations. Like the animated images, video visualisation is sensitive to map zooming, with the MapLibre (2023e) example in Figure 5 showing a lot of

detail in a relatively small area (the inverse of the animated images example).

A Web Map Service (WMS) is a standard protocol developed for serving georeferenced map images over a web browser (Figure 6). The WMS visualisation type provides an interface for visualising external web maps and requires an external web map server to host content.



Figure 5. An example of video visualisation superimposed on a map (MapLibre, 2023e)

For the use cases of visualising shapefiles and EPANET simulations, GeoJSON is likely to be the most suitable solution, given the need to work with large amounts of geographic data and for ‘close-up’ detail.

3 VISUALISING SPATIAL DATA VIA SHAPEFILES

The aqua3 project used seven pilot use cases to test the developed technologies: Trieste (Italy), Paris (France),

Brussels (Belgium), Limesos (Cyprus), Thessaloniki (Greece) and Sofia (Bulgaria). The pilot sites were using shapefiles to store a wide range of visual and quantitative data relating to the operation of their water networks. Figure 7 shows some of the data, including regional flood maps, areas of commercial importance, sites of wells, regional boundaries and so on.

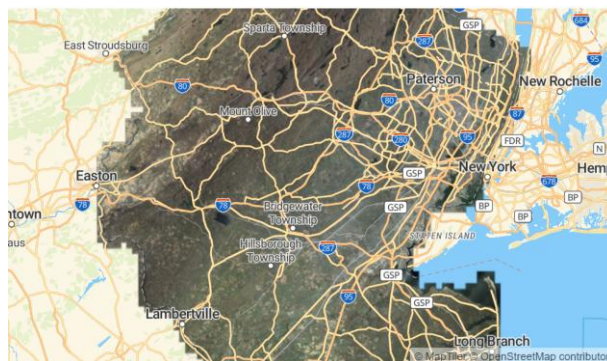


Figure 6. An example of the WMS visualisation with an added external raster layer map (MapLibre, 2023f).

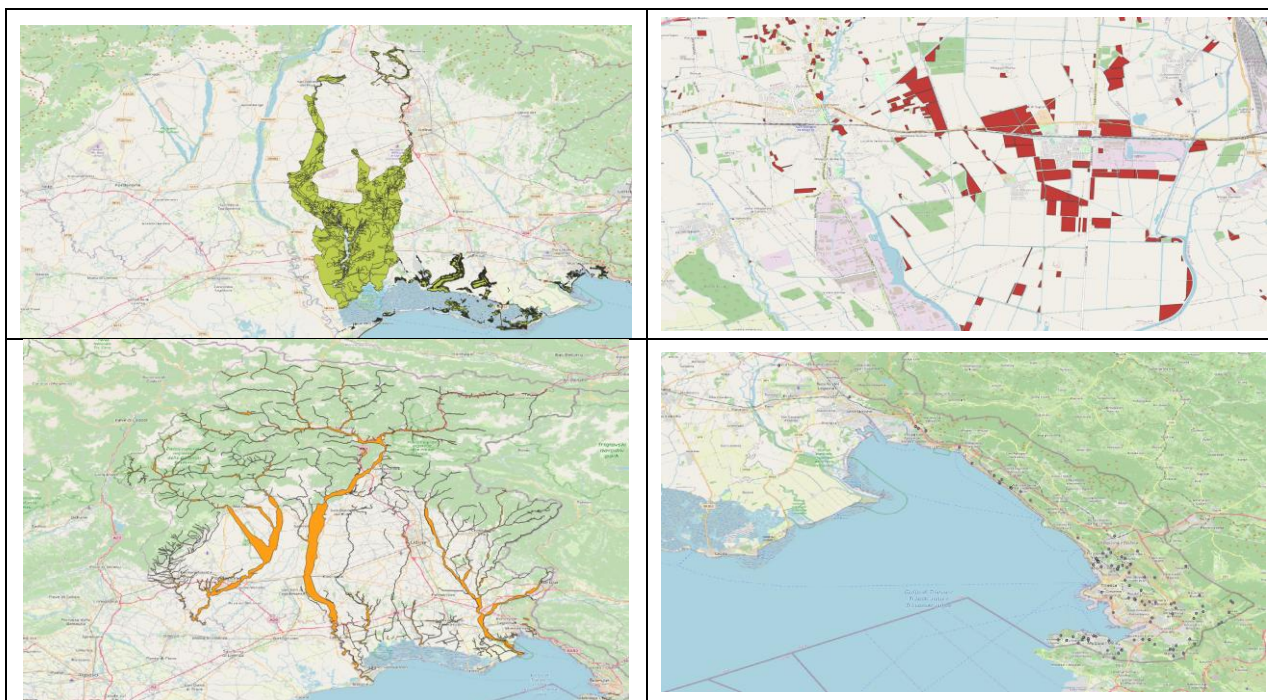


Figure 7: Typical shapefiles. Flood maps (top left), commercial locations (top right), river flows (bottom left) and water wells (bottom right)

The shapefiles presented for visualisation tended to be large and complex. Typically file sizes were in the order of 100-200 MB with geometry and metadata (providing information such as when and how data were gathered) for up to 150,000 features and associated metadata per file. The shapefiles covered up to an area of 60 km x 80 km with a resolution of 3 m x 3 m. Typical operation of the application would see all the shapefiles downloaded to the client, with users having the ability to show and hide different shapefile layers, and to reveal metadata associated with shapefile features.

Shapefiles are not a native data type for MapLibre, so there is no direct import function for them and they need to be converted into GeoJSON. That process could be undertaken on the client or server, depending on processing capabilities and project requirements. The larger shapefiles, containing more than 150,000 features, were likely to have a significant conversion load. Therefore, they were likely to create substantially large GeoJSON files which may not work well with MapLibre, particularly given the noticeable delays when working with the shapefiles in QGIS. Therefore a different approach was needed.

3.1 Approach

Initially, shapefiles were downloaded to the map-tile browser client and converted to GeoJSON in situ. This tended to be a time-consuming process that provided for a poor user experience. It also proved to be beyond the visualisation capacity of Leaflet.js, resulting in a move to first Mapbox and then MapLibre given its open-source credentials.

An offline approach was taken to convert shapefile data into GeoJSON using Geopandas (2023) and to transform the projection of the resultant geometry into WGS84 (The World Geodetic System 1984) format (the default for Mapbox). This was a slow process and undertaken infrequently, with the resulting GeoJSON files stored on a file server. Once downloaded to the client, GeoJSON data was loaded into the framework using `map.addSource` (MapLibre, 2023g) and `map.addLayer` (MapLibre, 2023h), as per the MapLibre sample. This resulted in a working 'proof of concept', i.e., it was possible to convert shapefiles into GeoJSON and have large GeoJSON files visualised in MapLibre with metadata available through mouse enter/exit events. However, the large file size of the GeoJSON

files (and number of files downloaded to the client) resulted in a long wait before all the GeoJSON layers were available to the user.

Visual inspection of the GeoJSON files revealed that around half of the file size was feature metadata with the remainder being feature geometry. The entire metadata was not required, as it was only referenced when the user's mouse entered a region and a pop-up dialogue displayed it. Therefore, it was concluded that the metadata could be downloaded on demand.

To further reduce the footprint of the GeoJSON data, the resolution of coordinate data was reduced. Geopandas appeared to default to 15 dp (post-decimal places) for WGS84 coordinate data which accounts for most of the resulting file size. Through trial and error, it was found that resolution could be reduced to 6 dp without 'noticeable' issues, though this is entirely dependent on the resolution of data that was being used. For example, it worked well for the 3m x 3m resolution of flood maps. It should also be noted that WGS84 measures in degrees, rather than 'absolute' meters. Therefore, the length of a degree, as meters, is a function of latitude, in that the Earth's circumference increases in size from pole to equator and is therefore not a consistent measure. The result of this is that our acceptable results for 34 degrees north may not be valid for other latitudes.

Finally, the resulting files were zipped, giving a footprint reduction from about 200 MB to around 10 MB. Whilst this data needs to be unpacked on the browser, it is a significantly less intensive operation than building the GeoJSON file on the browser.

3.2 Results

Figure 8 details the shapefiles from Figure 7 visualised as GeoJSON layers in the aqua3s client application. In general, MapLibre has proved to be a very capable tile-map platform, easily dealing with all the data that was downloaded onto it. One significant advantage in moving from Mapbox to Maplibre was the availability of source code which enabled us to solve a couple of troublesome issues with layer re-initialisation 'style.load' call-back (MapLibre, 2023i).

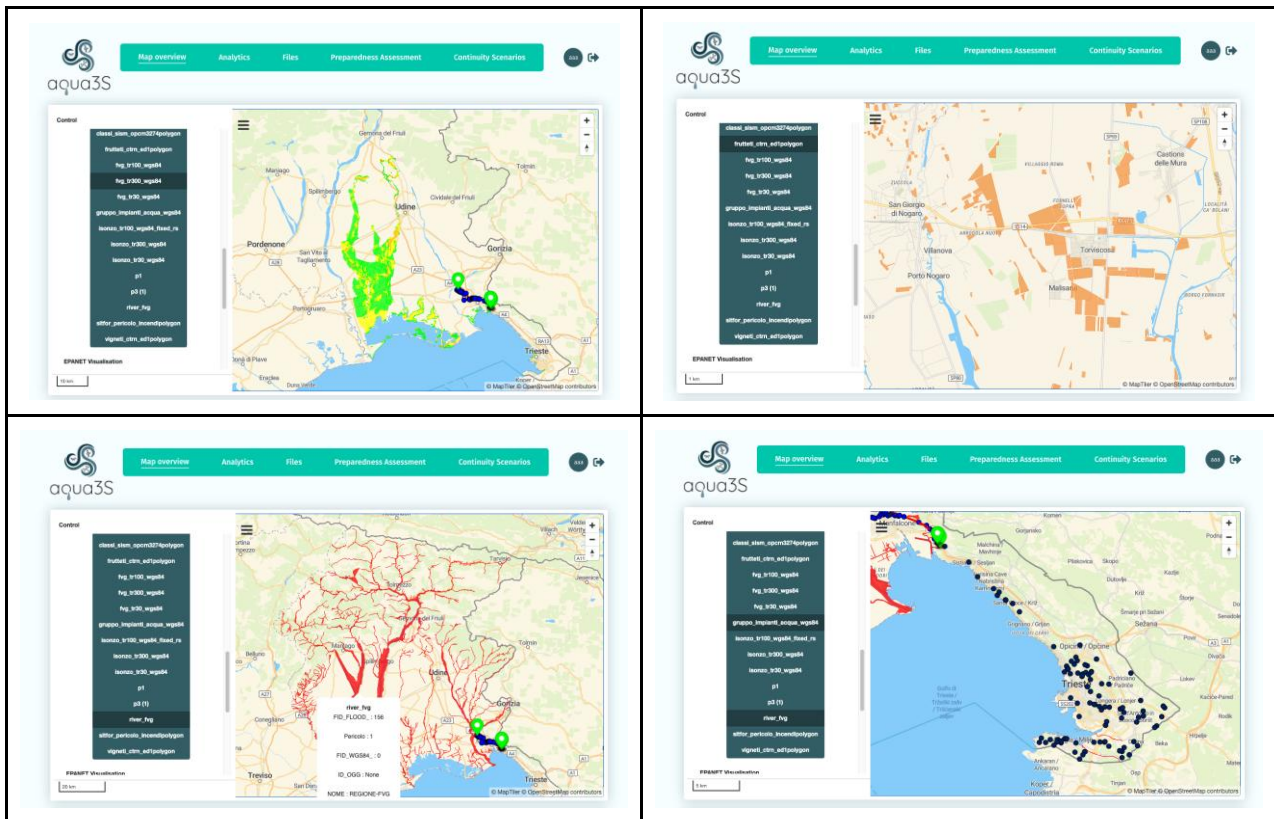


Figure 8: Shapefiles from Figure 7 visualised in aqua3s. Flood maps (top left), commercial locations (top right), river flows (bottom left) and wells (bottom right)

4 VISUALISING EPANET SIMULATIONS

For EPANET simulation, client users of the aqua3s platform wanted to be able to visualise water networks modelled with EPANET and view the results of hydraulic and quality simulations in a manner like the existing application (EPANET, 2023a), for example like in Figure 9.

4.1 Visualising Topology

On first inspection, EPANET networks appeared to be similar to shapefiles, in that they contain rich sets of geometry data that could be described as features. However, EPANET simulations give the network a dynamic quality, in the sense that the values associated with the network (e.g., pressures and flows) will change over time. The EPANET application displays this through colours and provides users with numeric data on demand. So, whilst the topology of the network does not change over time, its representation does. MapLibre

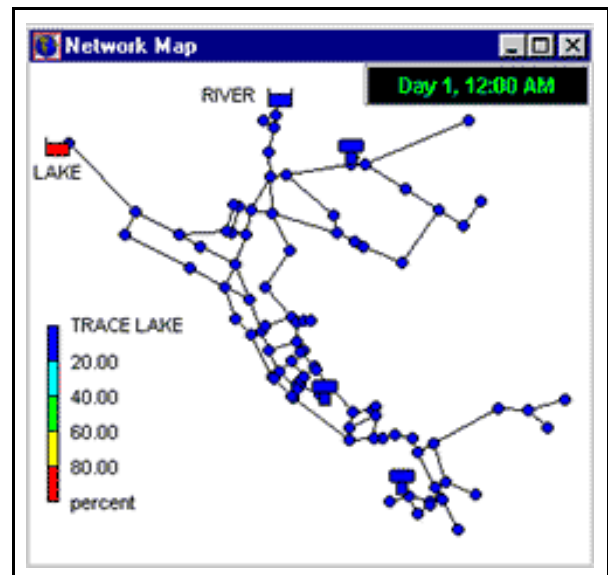


Figure 9: EPANET application water network visualisation

provides functionality to update layers in the setData() function, so an animated solution will need to build layer data (as GeoJSON), update layer data as the result of simulation data, and finally set the layer data to reflect the changes.

Constructing a GeoJSON representation of an EPANET network was a straightforward task, notwithstanding the need to spatially convert elements into WGS84, as required for MapLibre visualisation. The biggest challenge was in determining a suitable structure for the collection of EPANET UWDS component types which resulted in building a set of MapLibre layers to represent the pipes, pumps, valves, junctions, reservoirs,

and tanks. Whilst the MapLibre ‘add multiple geometries’ sample (MapLibre, 2023c) does show a single GeoJSON with multiple layers, we found that separating the data made it conceptually simpler to work with, particularly when dealing with simulation data and mouse enter/exit calls.

Having worked with large shapefiles of over 100,000 features, the typical water networks in Figure 10 appear small in comparison. Furthermore, given the relatively small payloads (about 1 MB) associated with these networks, there seemed to be little need to compress geometry data in the way that we had for shapefiles.

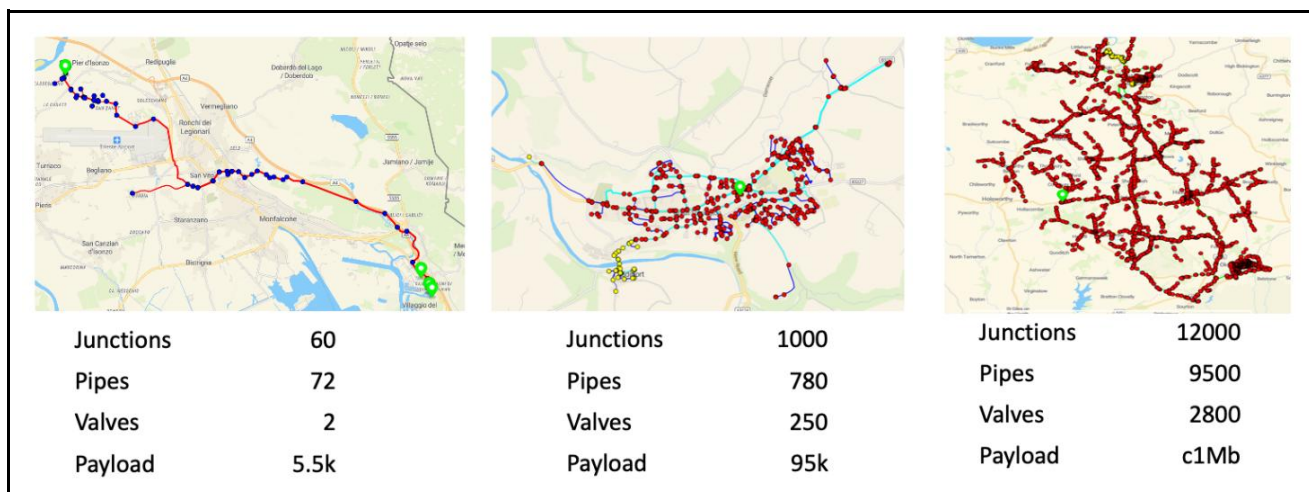


Figure 10: EPANET water networks of differing complexity

4.2 Simulation Data

A typical EPANET extended-period simulation runs over 7 days with a 15-minute resolution, creating around 700 data points per component per simulation. Simulations can be run in the EPANET app and saved as binary files (EPANET, 2023b) and/or run through the EPANET libraries, either to completion as per the app, or in a stepwise form. For our use cases, it was decided to run the simulation once and store the results for visualisation. Whilst the simulation was not massively computationally expensive, typically taking less than 2 minutes in the biggest network case, there seemed to be little value in re-running the simulation if users were unable to change simulation parameters.

For our large case, the resulting simulation output file was 100 MB. Whilst this could be loaded into memory

using Python and converted into Python data types, it was a slow process and echoed back to the shapefile metadata, in that not all users were going to require all of the data all the time. An approach was developed where the binary file was loaded, and data was unpacked on demand using Python’s struct.unpack_from (Python, 2023).

4.3 Simulation Process

EPANET divides components for simulation into one of two types, nodes, or links, and organises its colourisation visualisation around four modes for nodes (supply, head, pressure, and quality) and eight modes for links (flow, velocity, headloss, quality, status, setting, reaction rate and friction). Therefore, the simulation visualisation only needs to deal with a small subset of simulation data at any one time. However,

unlike shapefile visualisation where only user-selected metadata is downloaded, simulation visualisation requires all the node and link data to be downloaded for the given simulation frame and relevant modes. However, displaying this information visually, rather than quantitatively, requires colour information to be sent to the client which occupies a smaller data footprint, even more so when the visualisation uses EPANET's five colour lookup, Figure 9, effectively just requiring a map of indices to colour values and a colour index for each component. On the client, the simulation data was parsed and used to update the colour value of each feature in each layer of the model. On completion, setData was called and the visualisation was updated, Figure 11.

5 CONCLUSIONS

A client-server hydroinformatics model was created during the aqua3S project to provide GIS visualisation of EPANET modelling results. The results are presented within a web browser context, breaking down tasks between clients and servers that are linked by a computer network. The resulting software is easy to use by water management practitioners who do not have the software skills to develop such applications but require it for the design and operational analyses of water distribution networks. The application works on a desktop or laptop computer via its connection to the Internet. The process of creating the application and the tools used in the project are also described in detail to allow reproducibility of the research work.

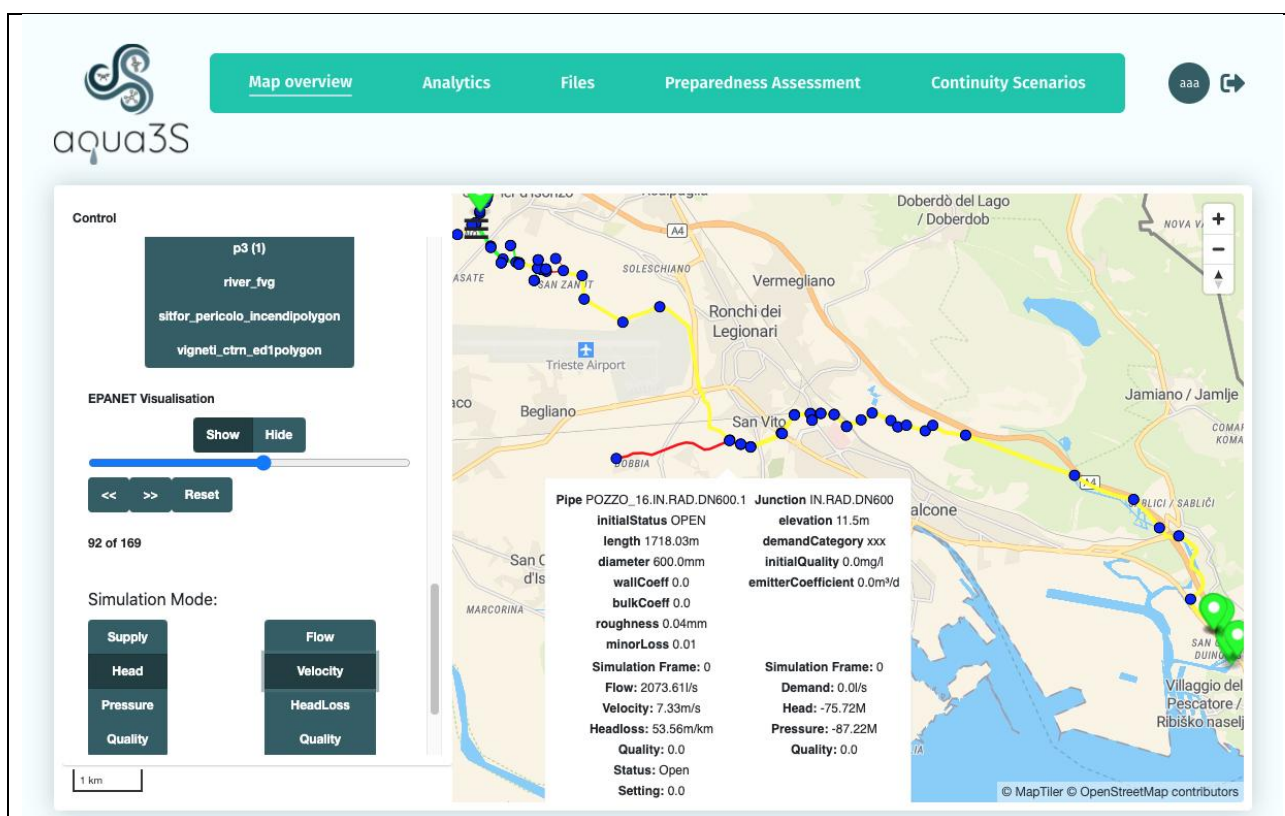


Figure 11. EPANET simulation visualization in aqua3s, showing coloured water network and simulation details

In general, using MapLibre as a tile map has worked exceptionally well. The library is generally easy to use and has a good level of visualisation performance. This has been tested and proved suitable for the size and complexity of data we have been working on the project with for both the EPANET and shapefile domains. The

open-source nature of the library enabled us to look into the underlying code when things were not working quite as expected and this did allow us to solve a particularly difficult challenges with refreshing visualisations after style.load.

To a large degree, choosing a suitable technology stack for the project was based on a trial-and-error process, in that it is a time and resource-limited activity. This may have resulted in a workable rather than an optimal solution. There is also the scope to re-engineer existing technologies to meet performance needs, particularly in the case of dismantling GeoJSON data to make it 'fit' our needs. Although GeoJSON seems to require large resources, it compresses well and keeps readability, which moving to a binary format would not allow so easily.

There are, however, some limitations to the current implementation. The most challenging issue is the time taken for the Python-based conversion of Shapefiles to GeoJSON. Future research will investigate the use of a different conversion library to improve the efficiency of the process.

ACKNOWLEDGEMENTS

The work presented in this paper was funded by the ongoing EC H2020 aqua3S (GA 832876), Fiware4Water (GA 821036) and LOTUS (GA 820881) projects, and the ongoing Water-Futures project which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (GA 951424).

REFERENCES

- [1] Makropoulos, C. and Savić, D.A., 2019. Urban hydroinformatics: Past, present and future. *Water*, 11(10), p.1959.
- [2] Maidment, D.R., 2002. *Arc Hydro: GIS for Water Resources*. ESRI Press, Redlands, California, USA.
- [3] Rossman, L.A., 2000. EPANET 2: Users Manual, US Environmental Protection Agency, USA.
- [4] Ahmadullah, R. and Dongshik, K., 2016. Designing of hydraulically balanced water distribution network based on GIS and EPANET. *International Journal of Advanced Computer Science and Applications*, 7(2).
- [5] Ramesh, H., Santhosh, L. and Jagadeesh, C.J., 2012, January. Simulation of hydraulic parameters in water distribution network using EPANET and GIS. In *International conference on ecological, environmental and biological sciences, Dubai* (pp. 350-353).
- [6] Bartolín, H., Martínez, F. and Cortés, J.A., 2008. Bringing up to date WDS models by querying. An EPANET-based GIS geodatabase. In *Water Distribution Systems Analysis Symposium 2006* (pp. 1-17).
- [7] Bayer, T., Ames, D.P. and Cleveland, T.G., 2021. Design and development of a web-based EPANET model catalogue and execution environment. *Annals of GIS*, 27(3), pp.247-260.
- [8] QGIS Home Page, 2023. <https://qgis.org/en/site/> (accessed 30 Sept 2023).
- [9] Aqua3s Project Homepage, 2023. <https://aqua3s.eu/> (accessed 30 Sept 2023).
- [10] Savic, D.A. and Banyard, J.K. eds., 2011. *Water Distribution Systems*. Thomas Telford Ltd, UK.
- [11] Google Maps Homepage, 2023. <https://mapsplatform.google.com/> (accessed 30 Sept 2023).
- [12] Mapbox Homepage, 2023. <https://www.mapbox.com/> (accessed 30 Sept 2023).
- [13] Leaflet.js Homepage, 2023. <https://leafletjs.com/> (accessed 30 Sept 2023).
- [14] MapLibre Homepage, 2023a. <https://maplibre.org/> (accessed 30 Sept 2023).
- [15] MapLibre - Add a default marker, 2023b. <https://maplibre.org/maplibre-gl-js-docs/example/add-a-marker/> (accessed 30 Sept 2023).
- [16] MapLibre - Add multiple geometries from one GeoJSON source, 2023c. <https://maplibre.org/maplibre-gl-js-docs/example/multiple-geometries/> (accessed 30 Sept 2023).
- [17] MapLibre - Animate a series of images, 2023d. <https://maplibre.org/maplibre-gl-js-docs/example/animate-images/> (accessed 30 Sept 2023).
- [18] MapLibre - Add a video, 2023e. <https://maplibre.org/maplibre-gl-js-docs/example/video-on-a-map/> (accessed 30 Sept 2023).
- [19] MapLibre - Add a WMS Source, 2023f. <https://maplibre.org/maplibre-gl-js-docs/example/wms/> (accessed 30 Sept 2023).
- [20] MapLibre - AddSource, 2023g. <https://maplibre.org/maplibre-gl-js-docs/api/map/#map#addsource> (accessed 30 Sept 2023).

- [21] Maplibre - Map.AddLayer, 2023h. <https://maplibre.org/maplibre-gl-js-docs/api/map/#map#addlayer> (accessed 30 Sept 2023).
- [22] Maplibre - MapLibre.map.on, 2023i. <https://maplibre.org/maplibre-gl-js-docs/api/map/#map#on> (accessed 30 Sept 2023).
- [23] EPANET Application, 2023a. <https://www.epa.gov/water-research/epanet> (accessed 30 Sept 2023).
- [24] EPANET Toolkit – Outfile Format, 2023b. http://wateranalytics.org/EPANET/_out_file.html (accessed Apr. 01, 2022).
- [25] Python, 2023. <https://docs.python.org/3/library/struct.html> (accessed 30 Sept 2023).
- [26] Geopandas Homepage, 2023. <https://geopandas.org/en/stable/> (accessed 30 Sept 2023).
- [27] Bing maps tile system, 2023. <https://learn.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system> (accessed 30 Sept 2023).

MODELIRANJE SISTEMA ZA DISTRIBUCIJU VODE U CLOUD-U: PRIMENA WEB-GIS ALATA

by

Gareth LEWIS¹, Bret SNIDER¹, Lydia VAMVAKERIDOU-LYROUDIA²,
Albert S. CHEN¹, Slobodan DJORDJEVIĆ¹, Dragan A. SAVIĆ²

¹ Centre for Water Systems, University of Exeter. Harrison Building, North Park Road, Exeter

² KWR Water Research Institute. Groningenhaven 7, the Netherlands

Rezime

Hidroinformatički alati se obično koriste za modeliranje i upravljanje vodoprivrednim sistemima. Te aktivnosti uključuju korišćenje različitih prostornih i vremenskih informacija prikupljenih iz različitih izvora i često dostupnih u različitim formatima. Razvoj ovih alata je često u domenu specijalizovanih programera i njihova efikasna upotreba od strane vodoprivrednih stručnjaka je veoma zahtevna. Softver ili aplikacije otvorenog koda, kao što su geografski informacioni sistemi (npr. QGIS) i softver za modeliranje sistema distribucije vode (npr. EPANET), su razvijeni da podrže analize i donošenje odluka ali se teško mogu spojiti da omoguće zajedničku primenu. Zato ova rešenja imaju neke kritične slabosti: 1) korisnicima može biti potrebno dosta vremena da postanu vešti u korišćenju date aplikacije, 2) aplikacije generalno pretpostavljaju određeni nivo znanja specifičnog za taj domen, 3) aplikacije mogu zahtevati prilagođavanje putem 'plugin' dodataka da bi pružili korisne informacije i 4) aplikacije su često vezane za specifične konfiguracije hardvera ili operativnog sistema. Da bi se pozabavili ovim problemima, istraživački projekat aqua3s koji je

finansirala Evropska Zajednica imao je za cilj razvoj cloud platforme koja integriše mogućnosti vizuelizacije i modeliranja koje pružaju QGIS i EPANET, koristeći cloud/internet za obezbeđenje funkcionalnosti.

Ovaj hidroinformatički pristup je rezultirao metodama koje mogu da spoje i vizuelizuju informacije potrebne za podršku upravljanju vodoprivrednim sistemima, posebno razvojem EPANET simulacije i vizuelizacije rezultata analize velikih vodovodnih mreža (preko 30.000 mrežnih čvorova i veza) putem interneta. Rezultirajuća aplikacija pokazuje kako se podaci mogu obraditi i uravnotežiti korišćenje klijenta i servera da bi se minimiziralo opterećenje klijenta i maksimizirala brzina odziva. Proces razvoja aplikacije i alati koji su korišćeni u tu svrhu u ovom projektu su takođe detaljno opisani kako bi se omogućila njihova upotreba u drugim istraživačkim radovima i primena u praksi.

Ključne reči: Hidroinformatika, open source, Web-GIS, EPANET, GIS, Shapefiles, Vizualizacija