CHEMICAL ENGINEERING TRANSACTIONS
VOL. 107, 2023
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.cetjournal.it
Guest Editors: Petar S. Varbanov, Bohong Wang, Petro Kapustenko

ISBN 979-12-81206-07-6; **ISSN** 2283-9216
DOI: 10.3303/CET23107010

Web-Based Map Service Connected to ROS2 Robot Network

Ferenc Péter Speiser

University of Győr, Department of Power Electronics and E-Drives, 9026 Győr, Egyetem tér 1., Hungary
speiser.ferenc.peter@ga.sze.hu

The key for testing the driving assistance and autonomous vehicle systems is to establish an advanced infrastructure. For special manoeuvres, a special track is necessary. The path of the track is always changing as the investigated function changes. This means a lot of track construction work for the operation of the proving ground. The engineers of the ZalaZone proving ground perform continuous track construction tasks for the various dynamic tests, which is a time-consuming, resource-intensive task. The goal is to develop an environment that can manage a semi-automatic test track building in a sustainable way. The placing of traffic cones can also be carried out by autonomous robots designed for this purpose, according to pre-designed plans, which are under development at the University of Győr. Automating this task can save a lot of energy, live work, and fuel also can optimize the load and the route of the robots. This paper discusses the theoretical design and possible physical architecture of a framework for the coordination of the tools used. It presents a possible software environment for managing and publishing data in a form that can be integrated with map data represented in different map projection systems. This requires the development of an HTTP-ROS2 interface that can transfer position and other descriptive data of participants communicating on the ROS2 network to the web server that provides the base maps. This work presents a map interface for the near real-time display of positional data from robotic tests carried out in the ZalaZone area. As a result, the map allows the movement of registered objects to be tracked and displayed on the web map via the internet on a publicly accessible server running on the university network. The server is currently capable of mapping the data recorded on the ROS2 network, but only on the local network. It implements the display of pre-saved data from the ROS2 network via the Internet. The long-term goal is to develop a web-accessible mapping framework that allows the current position of the cone-positioning robots to be displayed and controlled through this interface.

1. Introduction

In many cases, the use of spatial data and information to solve problems opens up new possibilities for evaluating outputs. Much of the information managed in databases has a spatial extent. The use of this dimension makes otherwise imperceptible connections clear in most cases. One of the earliest known examples of using geographic information to solve problems is a health epidemic problem. John Snow used a dot map that showed disease clusters of the cholera epidemic in London in 1854 (Tulchinsky, 2018).
Nowadays, it is hard to find a field where the spatial correlation of data is not also discussed. Some areas in general but not exhaustive: agriculture (e.g., fertility analysis), environmental science (e.g., the spread of pollution (Mascherpa et al., 2023), social sciences (e.g., distribution of unemployment), epidemiology (e.g., the spread of infection in a given area (Ali et al., 2022), business intelligence (e.g., the spatial dependence of purchasing power) and many more examples. The primary and obvious use in the automotive industry is navigation. This task can be achieved by optimizing several parameters. The classic target is simply the shortest route between two points, the only criterion being the distance traveled in this case. If the amount of energy consumed on the road is also taken into account, then the consumption of the vehicle must be estimated in some way. To do this, it is necessary to know what kind of landforms the vehicle has to move over – this information can be obtained from a terrain model – and it is also useful to know the temperature and general weather conditions in the area. For electric cars, it may also be a question of maximizing the rate of recuperation and integrating charging network points into the route. The best-known geographic information services are web maps that display the results of our searches on a map, showing the position, distance, and navigation options

Paper Received: 01 April 2023; Revised: 30 May 2023; Accepted: 13 June 2023
Please cite this article as: Speiser F.P., 2023, Web-Based Map Service Connected to ROS2 Robot Network, Chemical Engineering Transactions, 107, 55-60 DOI:10.3303/CET23107010

instantly. The aim of a development project at the University of Győr is to develop a robot that can automatically deliver traffic cones to the target position and place them on the road (Hajdu et al., 2021). This is necessary because vehicle development tasks require the continuous construction of various temporary test tracks and the collection of the traffic cones that make them up after the tests have been carried out. This is a repetitive task that is time- and energy-intensive, and automating this type of problem can increase cost efficiency. The primary goal is to create a proof of concept for using an autonomous robot for track construction. That can help with operating the proving ground since it is easier to satisfy the different track requirements of the proving ground users. The aim of this work is to develop a map that can support the tasks of autonomous traffic cone-positioning robots. Automating track construction is important from the operational aspect of the proving ground. The track construction could be faster, cheaper, and more accurate if a robot does that rather than an operator. The first step is to display the position and current status of the robots on the test track using the web-map interface. In the future, it will also be used to define and manage the tasks of the units.

2. Object and methodology of research

The aim of the work is to implement a web-based mapping service to support the autonomous control of the ZalaZone traffic cone-positioning robot. This requires looking at what base data are available in the area. Next, the software environment must be defined to manage and publish the data in a form that can be integrated with the data represented in different map projection systems. To achieve these goals, the appropriate base data must first be produced. The testing area used for the development is the ZalaZone University test track. For this area, it is necessary to collect as much and as accurate spatial data as possible on the map server on which the service is based. These data can be adapted to different service needs. The primary task of the map server was to provide accurate base information (orthophoto of the test track) for the map interface and to compare the position of the autonomously moving robots to an accurate reference. In addition, the map server can also serve other functions (e.g., route planning or road surface quality based on position). However, this information must be accompanied by appropriate base maps. The web map allows the movement of registered objects to be tracked and displayed via the internet. The most widely used framework for robot control is ROS2 (Robot Operating System 2) (Macenski et al., 2022). ROS2 is a set of software libraries and tools that help you create robotic applications. From drivers to state-of-the-art algorithms and powerful development tools, ROS2 has the functionality needed to control autonomous robots. ROS2 Foxy distribution is used in this implementation, and the user interface needs to be connected to it. This allows the units in the network to communicate in a pre-planned way. In order to display the information of the robots involved in the execution of the tasks in a web system, an interface needs to be developed. This interface can transmit position and other descriptive data of participants communicating on a ROS2 network to the web server running the map server at a suitable rate.

2.1 Base maps for the spatial services of the university test track

In terms of the data available, it is important to use a base map that is accurate for the purpose. The operating area of the robot is the ZalaZone test track, and the expected average traffic cone positioning accuracy is approximately 20 cm (this value can be improved). Based on the onboard RTK GPS, it is possible to determine the position with an accuracy of less than centimeters. The aerial footage available is a high-precision drone aerial image, which has been post-processed, as shown in Figure 1. The aerial image is in UTM33N projection with a resolution of about 5 cm/pixel. This map can also be used to generate additional derived data.

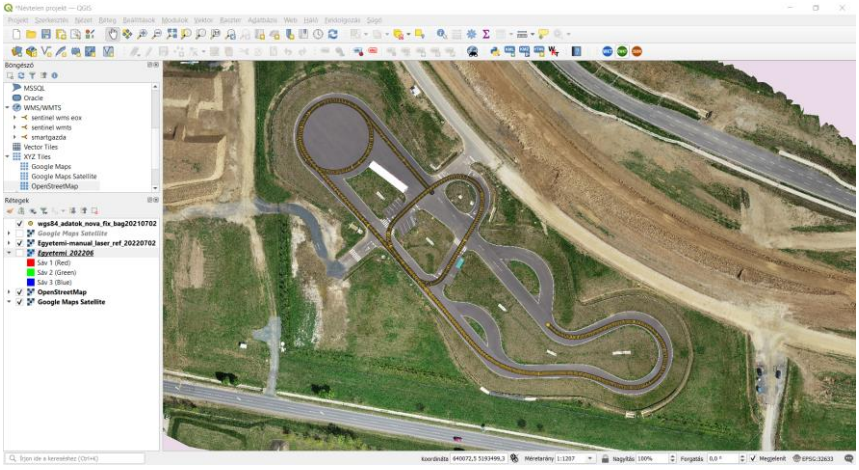

Figure 1: Orthophoto map superimposed on the ZalaZone university track area in Quantum GIS

The first need that arises is to determine whether a given point is on a road surface or not? The raster classification of the aerial photograph can be used for this task, but it is easier to edit the vector polygons of the road surface based on the size of the area using geospatial software.
Road data for the ZalaZone University test track are available on the OpenStreetMap server. After downloading these, it was found that they were not accurate, so the centerlines had to be corrected based on the available accurate orthophoto. After the operation, a line-type data layer in GeoJSON format was available, containing the centerlines of the routes, as shown in Figure 2a. From these road widths, an approximate road surface can be generated by zone generation. Distinguish between the pavement of parking areas, the asphalted surface, and the slippery section of the road as shown in Figure 2b. These data will allow the development of additional services on the university test track.

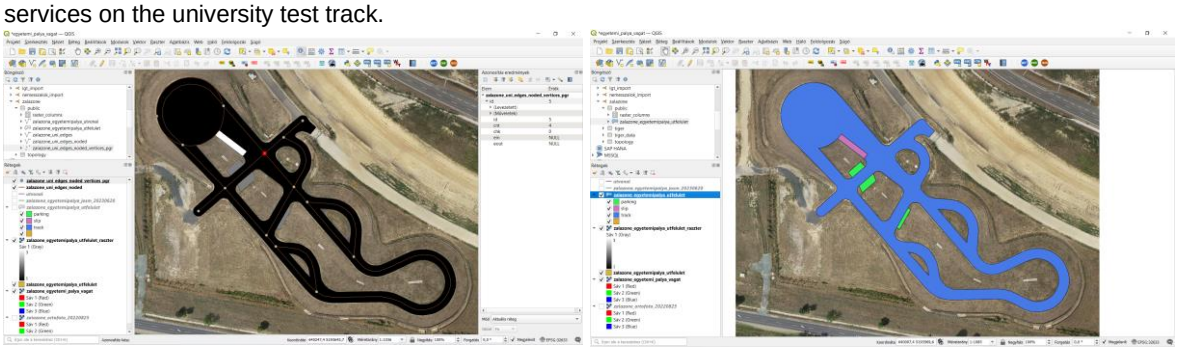

Figure 2: ZalaZone university track area a) route graph and b) road surface polygons in Quantum GIS

2.2 GeoServer – spatial data service

GeoServer is an open-source server for processing and publishing spatial data stored in various standard formats. The Java-based software can deliver OGC standard Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), and Web Processing Service on both vector and raster data. The spatial data is stored in a PostgreSQL database and a PostGIS extension. Services that can be used in the planned application:

- Orthophoto of the ZalaZone area (WMS)
- Land coverage data for the university test track area (WFS)
- Route planning for the university test track area (WFS)

2.3 User interface – front-end side technologies

The Leaflet JavaScript framework is used to develop the web map system. Of the map projections, WGS84, Web Mercator, and UTM are expected to be the most commonly used, and the EOV (Hungarian Standard Projection System) may also be considered. These can be handled by GeoServer in PostGIS and Leaflet as well. The common client-server architecture is used for storing the spatial data, as shown in Figure 3. The middle layer is responsible for the spatial data exchange between the client side and the database.

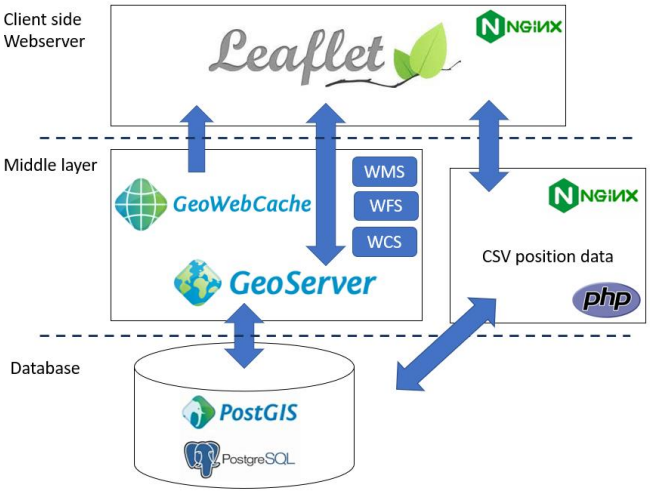

Figure 3: The relationship between the software implementing the map service

The user interface is responsible for displaying application data and communicating information to the user. The primary objective is to manage and display map and alphanumeric data related to the test area. Three main modules are identified: map, zoom, data and layer management.
Maps are managed through map layers in the system in the same way as in geographic information systems. The layer manager on the right of the screen allows users to select the base map to be used. The map can be used for the usual map operations: moving the map using the keyboard and mouse, zooming in and out, selecting a map object, and defining target positions. The application expects the incoming information in JSON format.
Simplicity should be the aim when designing the map interface. Apart from the map controls, all functions can be accessed via an icon bar. Functionality can be improved by expanding the toolbox. On the provider side, a Python server connected to the ROS2 network sends the information selected for display. From the client point of view, a WSS (Web Socket Server) connection must be implemented, which is implemented in a secure way in the browsers currently used. The client can currently send two commands to the server. One retrieves the positions of all active objects (get_positions), then processes the response and updates the position data of the objects on the map based on the incoming data. The other command is to send the target position of the selected object (set_positions) when the user clicks the send button on the map. It is important to note here that after the connection is initialized, the client requests data from the server at predefined intervals. With the current settings, this is implemented by the getDevicePositions() JavaScript function with a cycle time of 600 ms. The components of the planned system are connected as follows.

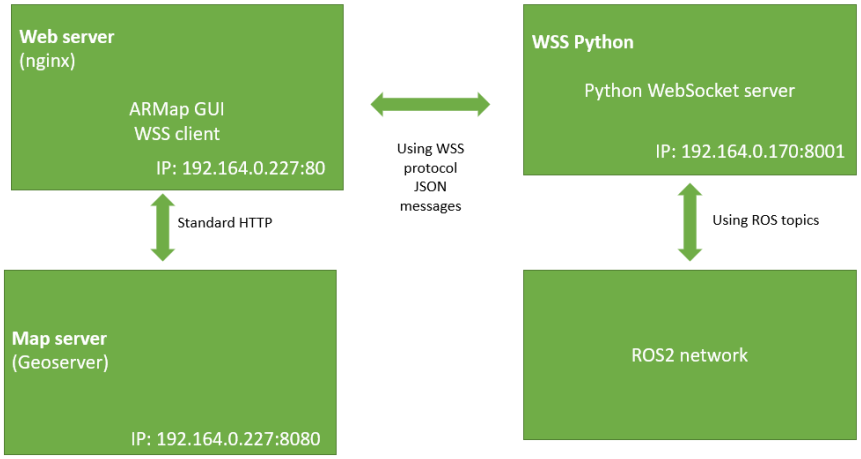

Figure 4: Structure of the prototype system

The application can work in two ways:

- The web server and the ROS2 gateway (WSS Python) are running on the same local network as shown in Figure 4. In this case, the components can run on a single computer, but the map is only available on the local network. (In this case, the corresponding port of the local web server can be published on a router accessible from the Internet via a fixed IP, but the same must be done with the communication port of the WebSocket server.)
- The web server is always available on the Internet, and the WebSocket server is located on a local network that is connected to the Internet and has a dedicated port on the WebSocket server that is accessible from the outside. Accordingly, when the web map is loaded, a duplex communication channel is initialized between the Python server and the web server so that the information flow is ensured.

The WSS Python server must perform the task of connecting the web server to the ROS2 network components. These components are devices that travel autonomously on the test track using their own internal data but communicate via the network and continuously provide data about themselves and receive control information. So the task is twofold:

- Selected information about objects on the ROS2 network must be transmitted to the web server and displayed on the web interface.
- On the other hand, the commands sent from the web interface (e.g., specify the target position of the selected device) must be delivered to the corresponding object in the ROS2 network for interpretation and execution.

The connections shown in the figure above should be implemented. In order to establish a permanent connection between the browser and the server registered on the ROS2 network, WebSocket technology is

used. WebSocket is a standard technology that implements full-duplex communication channels for web browsers over a TCP connection. Using this solution, the user interface can maintain a continuous connection to the ROS2 gateway. The gateway implementation requires a multi-threaded code to read the continuous ROS2 communication and send the information on the WebSocket side in parallel. This program thread is responsible for the implementation of communication with the outside world. It is responsible for receiving messages that come from the web application. The content of messages can be completely customized, implementing generic client-server communication. The client specifies a command to indicate what action it wants to perform on the server, and the server responds to the query in the appropriate format. Providing a communication channel starts with a handshake process. If there is an incoming request, the server receives it and sends feedback to the client side to open the channel. Then, the communication can begin. The channel remains open as long as the client session is maintained, so there can be a continuous flow of data between the front end and the back end. The physical channel and protocol are available, but it is also necessary to develop a message structure in which the read data can be transmitted in a way that is understood by the web server at the receiving end. For this purpose, messages in XML or JSON format are usually used for web development. After JSON conversion, the array of descriptive data generated from the ROS2 network messages is sent to the web viewer, where it is interpreted using JavaScript.

3. Results and discussion

The result of this work is a publicly accessible server running on the university network. The server is currently capable of mapping the data recorded on the ROS2 network, but only on the local network. It implements the display of pre-saved data from the ROS2 network via the Internet as shown in Figure 5.

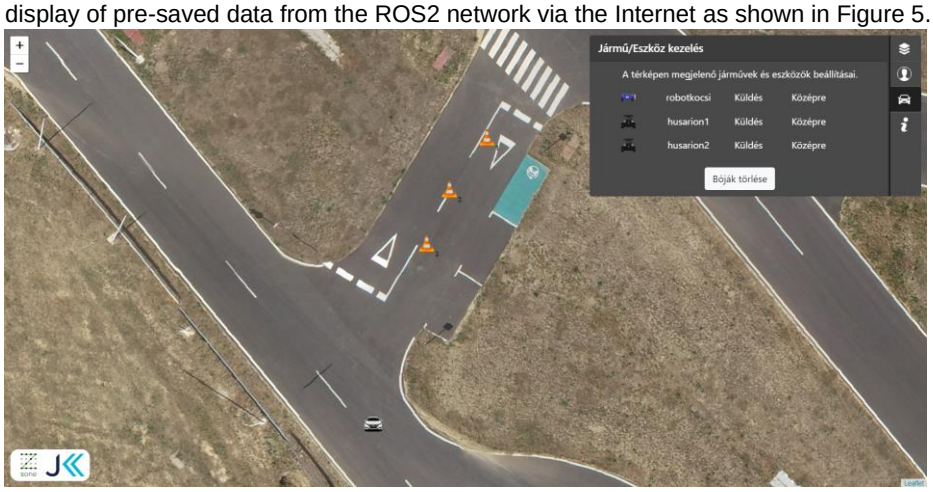

Figure 5: GUI for the map client

A further aim is to create a device that is easy to move, allowing easy portability and simple installation of the complete test environment. This can be done by installing each component on a portable computer, in this case a Raspberry Pi device.

When this device is connected to the network, all services are automatically available over the local network. The new needs related to the expansion of the spatial database services have generated the data content that allows the development of the necessary services. The design of the topology of the university test track will allow the definition of services that will allow robots to answer the spatial questions that are needed to implement autonomous functions. The questions are the following:

- what is the type of road surface at a given coordinate,
- what is the proposed shortest route between two points?

This raises the question of how the map can be used to solve the problem. The robot wants to know if the coordinate where it is or where it is instructed to go is on the asphalt. To do this, it sends a query to the map server by entering the coordinates and the metre-based coordinates of the UTM 33N in question. The second problem is the design of the route used to control autonomous robots. The user interface provides the possibility to enter coordinates and display diagnostic data for traffic cone placement. If they are done here, the basic data for the management tasks can also be entered by the central server. The navigation questions refer to the shortest recommended route between the coordinates within the course so that the route follows an asphalt surface as shown in Figure 6. To do this, a topologically correct road network map is needed that contains the

real paths of the real track. To solve the previous problem, we used a centreline map of the routes, and we will use this baseline data for the concept analysis.

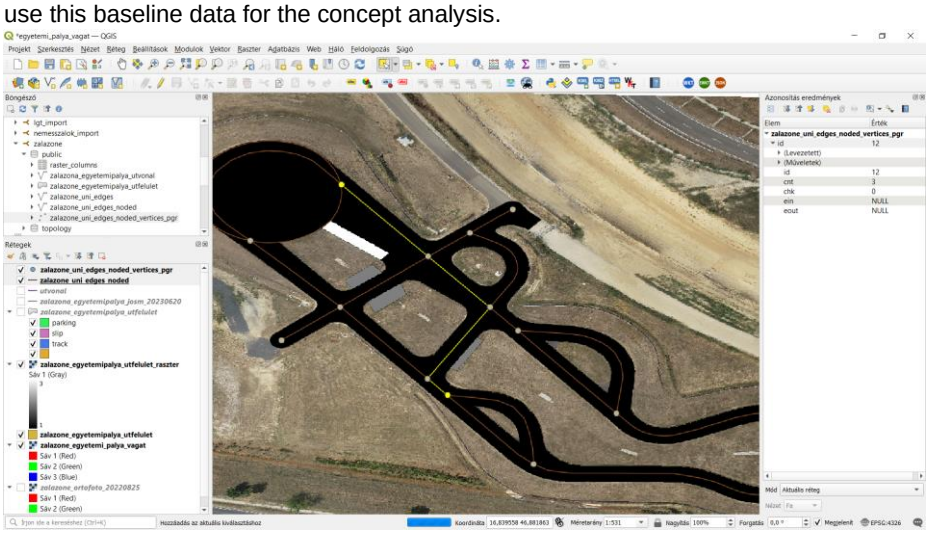

Figure 6: Route planning result between two selected points

4. Conclusion

The result of the work is a server and client application providing mapping services related to the operation of an autonomous traffic cone deployment robot, which is capable of tracking the movement of devices (robots) registered on the network, displaying diagnostic information, transmitting control information (sending the positions of traffic cones to be deployed). The implementation of a complex map server that can be easily adapted to any local network in the form of a mobile computer that can be easily moved was carried out. It has helped to simplify the preparation of field tests. By connecting the server to the network, the map services and user interface will also be available on the local network. Further work will be done to implement a web connection to the WebSocket server running on the ROS2 network in order to establish a direct connection to the devices registered on the ROS2 network. In this way, their data can be continuously managed on the map, allowing diagnostic data to be displayed at multiple levels of authorization. Once this information is available in the database, the optimization of asset movements and the optimal placement of selected traffic cone configurations can be planned. This will improve the efficiency and makes more sustainable the operation of the proving ground.

Acknowledgements

The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002).

References

Ali M., Moses A., Kweku N.E., Punguyire D., Spears N.C.B., Mawupemor A.P., Abdul B.K., 2022, Spatial epidemiology of bacterial meningitis in the Upper West Region of Ghana: Analysis of disease surveillance data 2018–2020. Clinical Infection in Practice, 16, 100160.

Hajdu Cs., Boros N., Ballagi Á., 2021, Development of track-building mobile robot based on cognitive-inspired motion planning architecture, 12th IEEE International Conference on Cognitive Infocommunications Proceedings, 23-25 September, Online, ISBN 9781665424950, 781-786.

Macenski S., Foote T., Gerkey B., Lalancette C., Woodall W., 2022, Robot Operating System 2: Design, architecture, and uses in the wild., Science Robotics, 7(66), eabm6074, DOI:10.1126/scirobotics.abm6074.

Mascherpa M., Haasler I., Ahlgren B., Karlsson J., 2023, Estimating pollution spread in water networks as a Schrödinger bridge problem with partial information, European Journal of Control, DOI: 10.1016/j.ejcon.2023.100846.

Tulchinsky T.H., 2018, John Snow, Cholera, the Broad Street Pump; Waterborne Diseases Then and Now., Chapter 5, In: Case Studies in Public Health, Academic Press – Elsevier, London, United Kingdom, 77-99, DOI: 10.1016/B978-0-12-804571-8.00017-2.