Developing a tool for assisting non GIS experts in the automated creation of ready-to-print maps

Roland Lukesch [a,*], Lukas Nebel [a]

[a] *Verkehrsverbund Ost-Region (VOR) GmbH / ITS Vienna Region, Austria – roland.lukesch@its-viennaregion.at, lukas.nebel@its-viennaregion.at*

* Corresponding author

Keywords: automated map creation, transit maps, public transport, transport mapping, infrastructure

Abstract:

Within the public transport authority of Burgenland, Lower Austria and Vienna (Verkehrsverbund Ost-Region GmbH VOR), there is a regular need for static, ready-to-print map products with recurring characteristics, but variations in detail. Recurring are the types of the requested maps, like the routes of individual public transport lines, local areas around bus stops, or territorial units covered by the public transport network. Variations come, for instance, from the exact definition, which municipality or public transport line is to be depicted, or whether the map is to be enriched with additional information such as areas with high population density. The maps are needed for the information of employees, for internal presentations, or for communication with owner representatives and the public. This results in high demands on map layout and design, which are not met with existing tools for interactive mapping. Until recently, these maps are created individually by employees of the GIS department. After finishing the project described in this abstract, this process has been automated in order to relieve GIS staff and provide a faster service to internal users. The idea is that users are now able to specify the desired map type and corresponding parameters via a web frontend, initiate a backend process for map creation, and in turn download the completed map export via the frontend. The aim of the system architecture is to use the rich possibilities of GIS software for cartographic visualisation together with an easy to use dynamic web frontend to support non GIS expert users in creating appealing maps according to their (most common) needs. At the beginning of the project, it was determined to use open source software for the implementation. Therefore, our initial research focused on the available tools rather than the scientific literature. The components of the system to be developed should also be made publicly available as far as possible. This resulted in the Python package *autoMaps*.

The development of the software, the adaptation of the underlying database and the specific configuration were done in close coordination with the future users. The resulting map service went live within the VOR under the name *MapsToGo*.

The data on which the maps are based (territorial units, public transport lines, stops, etc.) are imported from different public or internal source systems into a PostgreSQL/PostGIS database. Import scripts ensure consistency and repeatability, the definition of indices short processing times. Given the wide range of possible combinations of map content, the modelling of the basic data is important for the quality of the outcome. To achieve this goal, each map type also has its predefined set of combinations of map elements.

QGIS is used for map generation, based on generic layouts, which act as templates. The wide range of scales and densities of map content, varying greatly depending on user input, presented a particular challenge. The use of the recently available basemap.at vector tiles by geoland.at (the official geodata provider of the Austrian administration) and the resulting possibility of fine-tuning the content and display of the integrated base map proved to be extremely helpful. This way the basemap is changed in colouring, and some elements are combined. It's also possible to use a version without any labels. The creation process is controlled via pyQGIS, QGIS' Python API. This process typically involves setting layer visibility, defining filter expressions or the map extent, setting texts such as headings or source information, and, finally, the export as PDF, PNG or SVG file. Figure 1 shows an export created by *MapsToGo*.

Users can set the map type and its parameters via a web frontend, which was implemented using the Python package Streamlit (https://streamlit.io/). During this process, database queries are continuously performed to populate the interactive elements of the frontend (e.g., selection lists).

To make the system multi-user capable and to enable asynchronous generation of maps, a messaging architecture consisting of frontend processes, a central registry and several worker processes was designed and implemented using the messaging library ZeroMQ (https://zeromq.org/). The workers can process map requests independently of each other in

Abstracts of the International Cartographic Association, 5, 2022.
European Cartographic Conference – EuroCarto 2022, 19–21 September 2022, TU Wien, Vienna, Austria.
https://doi.org/10.5194/ica-abs-5-129-2022 | © Author(s) 2022. CC BY 4.0 License.

parallel. The registry keeps track of the available workers and their states, thus ensuring efficient allocation of tasks. An overview of how the described components work together can be found at Figure 2.

Throughout the software development process, great emphasis was placed on decoupling configuration (e.g., UI elements, processing steps, logging settings) and program logic. The resulting *autoMaps* package is thus easily reusable for other projects, whether internally or in other organizations. Great importance was also attached to the development of comprehensible documentation, a coherent command line interface and the possibility of creating sample projects. The system has been configured for internal use and installed on two Ubuntu servers.

The first version has met with good feedback from the user group. The *autoMaps* code repository is available on GitHub (https://github.com/itsviennaregion/automaps). It is our goal to attract new contributors to the *autoMaps* package and thus establish a jointly supported open source project. The project team is convinced that *autoMaps* can be successfully used by other organizations to create maps more efficiently and hopes for more deployments of *autoMaps* in the future.

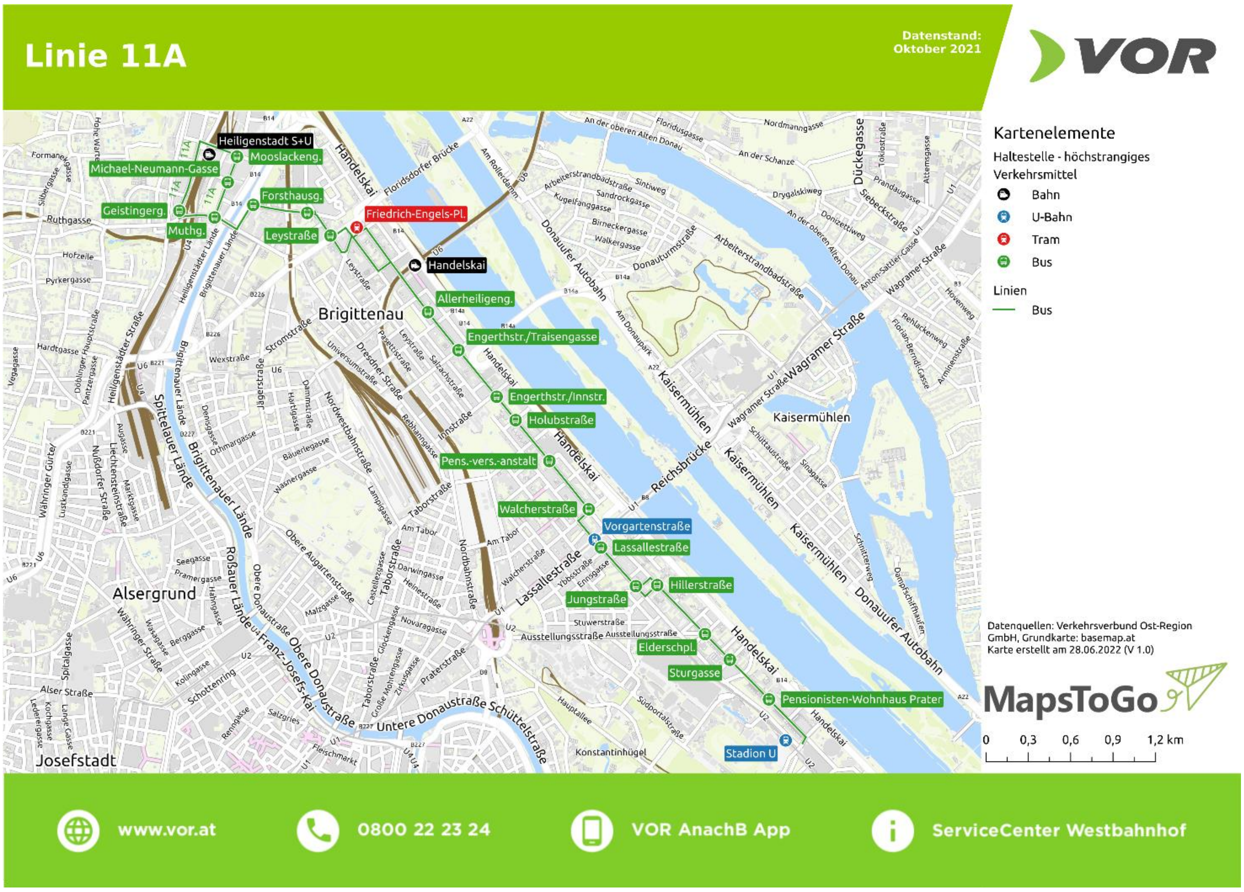

Figure 1: Example of a map output. Users can choose between various scopes, map elements and background maps. In this case the route of the line 11A and its stops are displayed on top of a basemap.at background map.

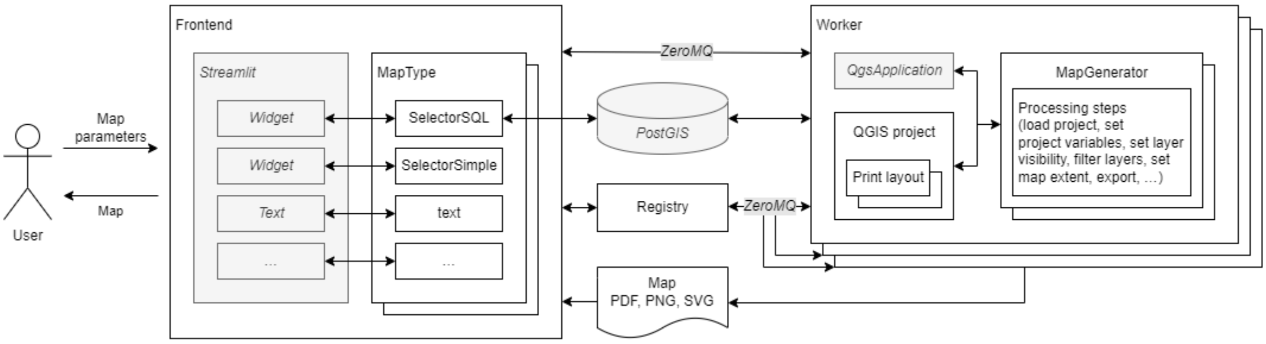

Figure 2: Main components of an autoMaps project. Existing standard software is greyed out.

https://doi.org/10.5194/ica-abs-5-129-2022 | © Author(s) 2022. CC BY 4.0 License.