Cloud-based software tools for the rapid development of web GIS

Maxim I. Malimonov[1], Oleg E. Yakubailik[2]

[1]*Federal Research Center Krasnoyarsk Science Center SB RAS, Krasnoyarsk, Russia*
[2]*Institute of Computational Modelling SB RAS, Krasnoyarsk, Russia*

Abstract

The article discusses methods for building geospatial web applications based on the ICM SB RAS geoportal resources. The sources of information for them are two types of data: web-service-based cartographic layers and data from the operational monitoring sub-system. The developed instrumental software allows you to select data on the geoportal, form, and configure the user interface. The modular structure and functional capabilities of the developed system are considered.

Keywords

JavaScript, JSON, vue.js, Web GIS, geospatial data, web mapping, thematic layer, geoportal.

1. Introduction

With the development of web technologies, ordinary users can work with various information, using programs and web applications as tools for analysis and viewing. Basically, such an application is a web browser, which allows you to use information systems without pre-installation. The web environment serves as an excellent basis for development, having many software tools in its field that allow you to create user interfaces, components, and applications.

Despite the variety, there are few solutions for the rapid development of geoinformation web systems. One of them is the ArcGIS Web AppBuilder, which allows you to download various resources, interact with them using a wide library of tools, configure the user interface, change the layer's attribute information, etc. It should also be noted NASA products, for example, Worldview or TerriaJS from Terria and many other geoportals [1, 2, 3]. These websites allow you to view the available ready-made resources, but, unfortunately, they can only be viewed and cannot be affected or changed in any way. In this regard, there is a need to develop a tool for deploying applied web GIS. This article demonstrates a system for building GIS-oriented applications in a web environment with a flexibly customizable user interface and providing various functionality: creating and editing projects, uploading and editing various resources, connecting customizable components for interacting with data.

Currently, the possibilities for visualization and analysis of operational environmental monitoring of air in the city of Krasnoyarsk using the developed software components have been

SDM-2021: All-Russian conference, August 24–27, 2021, Novosibirsk, Russia
malimonov.mi@ksc.krasn.ru (M. I. Malimonov); oleg@icm.krasn.ru (O. E. Yakubailik)
0000-0002-2668-4776 (O. E. Yakubailik)

implemented. This system is developed using the Node software platform.js, based on the V8 engine, an open source JavaScript framework Vue.js, MongoDB databases.

The sources of information are the resources of the geoportal of the Institute of Computational Modeling, Siberian Branch of Russian Academy of Sciences (ICM SB RAS) (http://gis.krasn.ru).

2. Software structure

The developed software consists of several modules. A module is a complete functional fragment of a program. It is designed as a separate file with the source code or a continuous part. A module consists of a single file containing its own logic, imported libraries, files or components, and blocks for defining HTML, CSS, and JavaScript.

There are three large modules in the system, which are divided into smaller parts. The first module is “Personal Account”. It is responsible for the user’s login and for editing user data. The second block is “Project Manager”. It allows the user to view existing projects, create new ones and edit information about their projects. The third module is the “Workspace”. It allows you to display various available resources from the ICM SB RAS geoportal on the map, change their display style, set different classes with the style and display conditions, etc. All modules are connected to two databases: the ICM SB RAS and the project repository.

Each parent module has several children. For all child modules, there are single layouts for display, depending on the parent module. There are static and dynamic parts, which are then displayed as a page. The “Personal Account” has its own layout, in which all child elements are already displayed, and the “Project Manager” also has its own layout. When changing the selected child module, the necessary data is transferred from the parent to the child, and the module is selected in the dynamic part of the layout. Figure 1 shows the process of forming a daughter module from dependent components.

The navigation menu is displayed in the static part of the layout. When you select a menu item, the dynamic part changes. The child module of the selected menu item is merged in it. Figure 2 shows the process of forming a page based on child modules for display.

There are two databases that the system communicates with. The first one is the database of the system itself. This database stores the created projects and copies the users who are part of

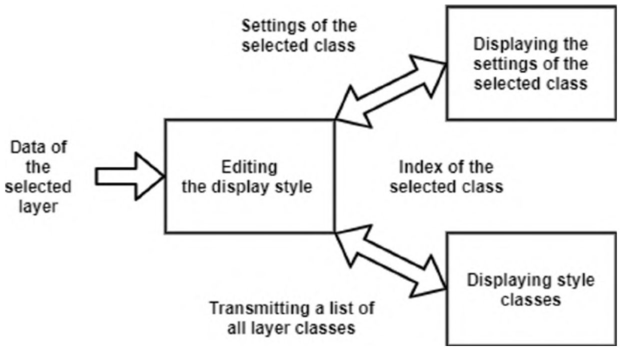

Settings of the selected class
Displaying the settings of the selected class
Data of the selected layer
Editing the display style
Index of the selected class
Displaying style classes
Transmitting a list of all layer classes

Figure 1: The scheme of formation of the child module “Editing the display style”.

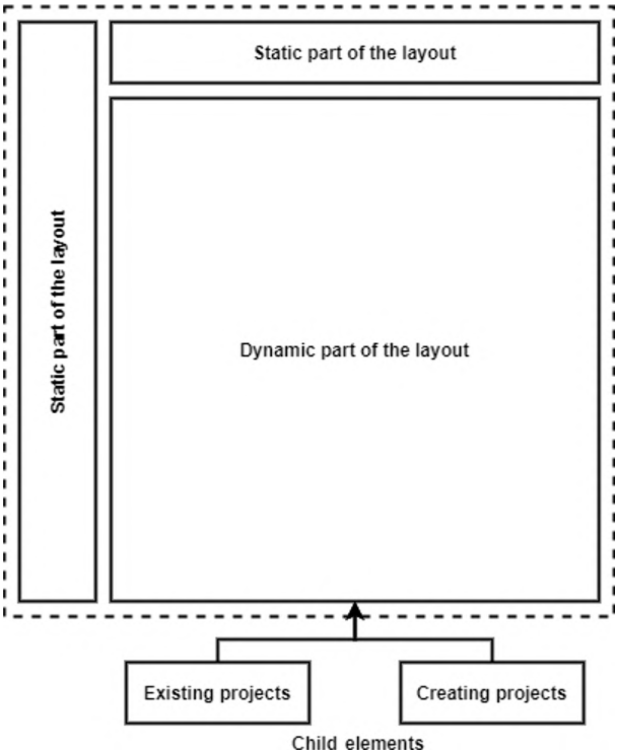

Static part of the layout
Static part of the layout
Dynamic part of the layout
Existing projects
Creating projects
Child elements

Figure 2: The scheme of forming the layout for displaying.

the ICM SB RAS geoportal to not create separate users for the geoportal and the application building system. The second is the database of the ICM SB RAS geoportal. It is a huge dataset, including data from the environmental monitoring network of the Krasnoyarsk Scientific Center. All interactions are carried out using the REST-based geoportal API [4]. Figure 3 shows the system's modular structure, the communication of the three functional modules with each other, and how they communicate between the two databases.

The MongoDB database is used to store system information for deploying application applications. MongoDB is a documented database management system that uses the JSON format for storing data. Two data collections have been created in the developed database. The first collection is responsible for storing information about users; the second one is for storing projects. The author's name is used to link users to their projects. It searches for owned projects and opens access to editing them.

The system works using Node, Vue, and the MongoDB database. Node is a software platform based on the V8 engine that allows you to use the JavaScript programming language as a general-purpose language. This means that JavaScript can be used to build web pages and interact with them and implement a server [5]. The web application itself cannot access the database and process various requests from the user. To do this, a server is used, to which various requests and user actions are transmitted from the web application.

Figure 4 shows the interaction of these three components. As shown in the figure, the web

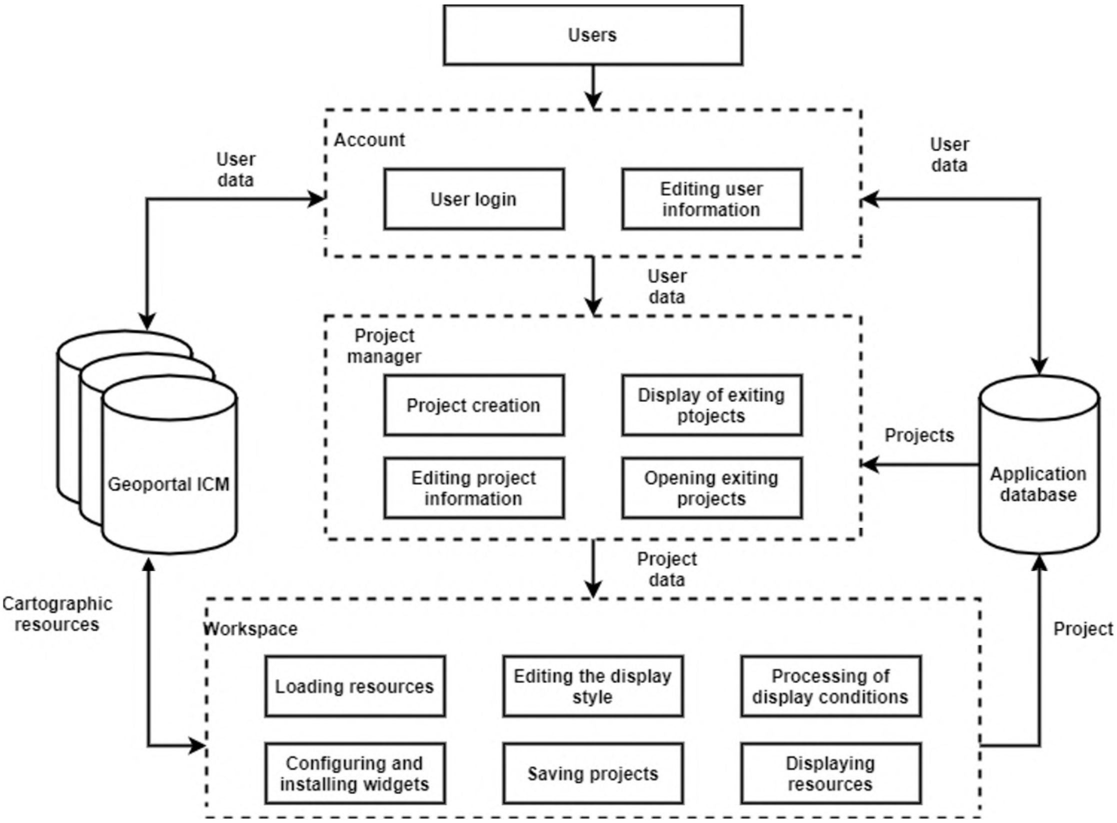


Figure 3: General system architecture.

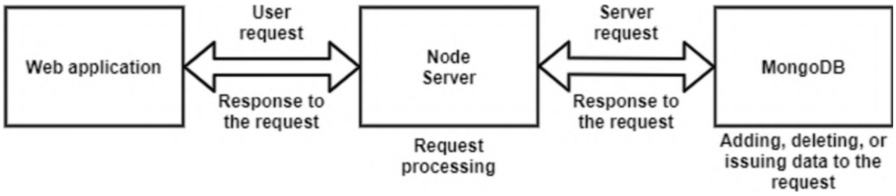


Figure 4: The scheme of the system’s work with the server and the database for storing the project.

application sends requests from the user to the server (Download the project, Save the project, Get project metadata, Get user data, etc.), then the server processes the requests and forms a request to the database. The response is returned along the chain to the user. Requests from the user are formed according to the architectural style of interaction between REST components that use HTTP requests.

The server can process the following requests and perform certain actions:

- Request for registration. The same login and password are used to access the geoportal of the ICM SB RAS and the system. If the login was successful, but you were not previously registered in the system, the system will notify you about this and create a new user record and a place in the database for storing projects. Then you will be redirected to a page where you will need to fill in information about the user.

- Login request. If the login is successful, and you have previously logged in, then you will be immediately redirected to the page with all the projects.
- Request for projects. The result of the request is a list of projects from the database.
- Save request. As a result of this request, the project is saved from the "Workspace" or from the "Project Manager".

The user interface is built on a JavaScript framework Vue.js. Inside this framework, all work and communication between components are carried out according to the MVVM template. This template allows you to separate the model and views to change them separately from each other. The MVVM template is divided into three parts:

- *Model* describes the data used in the application and may also contain various logic associated with this data. It is the foundation for the necessary operation of the application.
- *View* is a graphical interface. The view can be subscribed to data update events, property changes, etc. If the property has changed, this is reported to all subscribers, and the View requests the updated values from the Model.
- *View Mode* contains the Model converted to the View and commands that the View can use to influence the Model.

3. Developed software functionality

The system allows you to request various cartographic resources available on the ICM SB RAS geoportal. This includes resources related to administrative borders, water systems, operational monitoring, etc. There is a separate database of the ICM SB RAS, which stores operational environmental monitoring data. Various data from monitoring posts: wind speed and direction, temperature, humidity, pressure, PM2.5, AQI, MPC fractions, etc. Based on these data, we will consider the operation of the system.

Working with the system begins with the "Personal Account" module, where you need to enter your username and password. After logging in and receiving a key from the server to access various remote resources, the transfer is made to the "Project Manager" module. This module requests the database with projects through the server; the server returns a list of all available projects to the user. As mentioned earlier, now, there are only four child modules of the "Project Manager", three of which are displayed as a page, and the user can interact with them — this is the display of existing projects, editing, and creating projects. The fourth module works only if the user clicked "go" on the project card, and this module executes a request to get the project structure and transmits data to the "Workspace". At the moment, the module "Creating a project" is of interest. "Creating a project" — is a page with the initial parameters of the project:

- project name;
- project Description;
- initial substrate;

- camera coordinates;
- zoom in on the camera;
- project coordinate system.

After filling out the form, the user opens the "Workspace" module. This module is a page with two menu panels and a map. The first panel, attached at the top, displays the project name and a button to return to all projects. The second one, attached to the left, displays the active layers of the project and tools. When you click on "Layers", a submenu opens with the ability to add various cartographic resources from the ICM SB RAS geoportal.

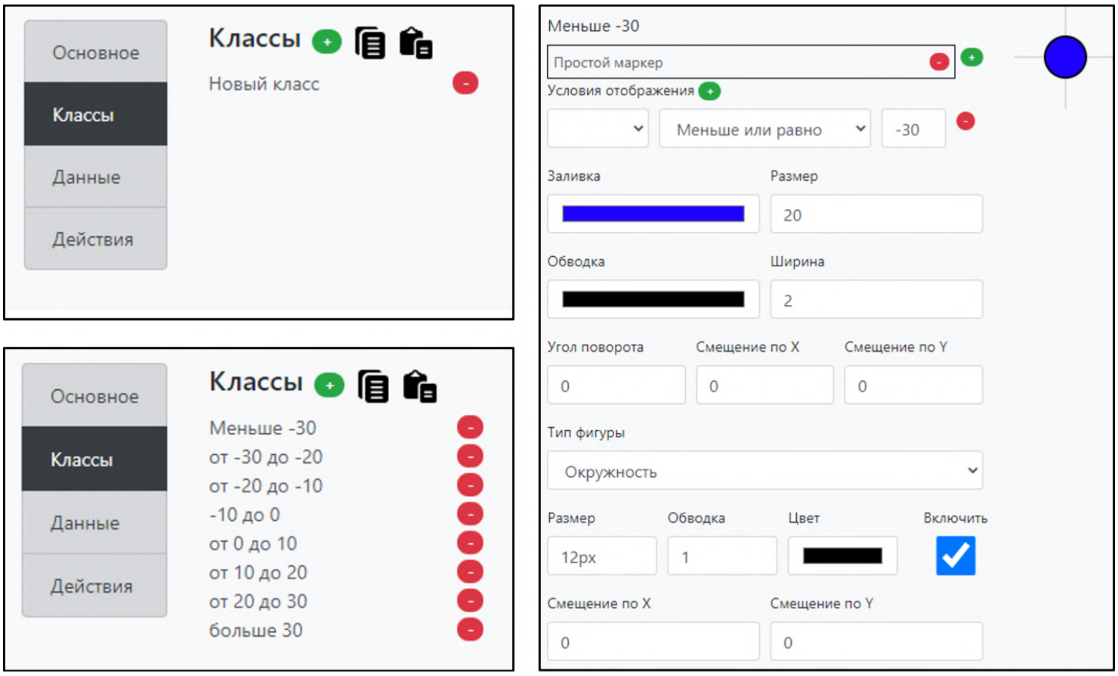
It should be noted that today there are many tools and ready-made systems that offer visualization services and analytical processing of operational monitoring data [6, 7, 8, 9, 10]. The features of the proposed implementation will be discussed below.

Let's consider the cartographic resources from the "Operational monitoring data" database and build a small project to display data from monitoring posts included in operational monitoring. For example, we will display temperature data for a month. In the beginning, you need to click on the "Layers" panel and click on the "+ Add layer" button. In the menu that appears, please select the posts that interest us and close the add menu. In this example, the monitoring posts "Peschanka", "Pokrovka", "Sverdlovsky", "Svetly", "Solontsy" will be used.

In the "Layers" menu, the selected monitoring posts will appear as separate layers, and when you click on such a layer, the settings menu will open. You can configure the layer's behavior in this menu, the display style (marker color, rotation angle, offset, etc.), classes, projection, etc. Let's look at the operation of the system using the example of temperature data. There is a tab "Data" in which all available data of the layer's monitoring post or tabular data is displayed. Select the item "Temperature". Such systems use coloring depending on the temperature. This coloring can be configured in the "Classes" menu in the settings and set the conditions for which values this class will be triggered.

Let's take the temperature range from -30 degrees to $+30$ and set the step to 10 degrees. You will need only 8 classes with their own coloring. Two classes will be responsible for temperatures outside this range. One class corresponds to a temperature that is less than -30. The other class corresponds to a temperature that is greater than $+30$. Let's call the classes names that will reflect the ranges. Figure 5 shows the class menu. After adding all the classes, you need to configure the display style of each class and set the trigger conditions. Clicking on the class opens a menu with display settings. In this menu, there are settings for different types of layers. Since we are currently working with monitoring posts, the settings for them are the same as for points. There is a small marker constructor that allows you to create composite markers. You can specify the fill and stroke color. Rotation angle, X and Y offset, shape type (circle, square, star, etc.), and text settings. If necessary, you can add several types of shapes to one marker. The interface for setting parameters is shown in the same Figure 5.

The next step is to download data from monitoring posts for a certain period. To do this, there is a "Tools" menu where you can select the "Calendar" tool. In the calendar, you need to set the download range and time step. Take, for example, the month of March in one-day increments. The data will be automatically loaded based on the added posts, and will be stored in the system until the page is reloaded. Now you need to add a timeline from the "Tools" menu, which will allow you to switch the selected days and update the information on the map.



Классы

Новый класс

Классы

Меньше -30
от -30 до -20
от -20 до -10
-10 до 0
от 0 до 10
от 10 до 20
от 20 до 30
больше 30

Меньше -30

Простой маркер

Условия отображения

Меньше или равно -30

Заливка

Размер 20

Обводка

Ширина 2

Угол поворота 0

Смещение по X 0

Смещение по Y 0

Тип фигуры Окружность

Размер	Обводка	Цвет	Включить
12px	1	■	☑

Смещение по X 0

Смещение по Y 0

Figure 5: The menu of the layer display classes and the settings of their display parameters.

By moving through the timeline, the layer calculation event is triggered. This event calls the calculation function, in which each layer is taken, and each class in the layer is viewed. All conditions in all classes are checked and compared with previously loaded values. After all this, the selected layers are converted to GeoJSON format, which the OpenLayers library can recognize, and passed to the "Resource Mapping" module. In this module, layers are iterated, and the main style class is applied to them, and the layer is displayed on the map (Figure 6).

In order to display all the classes of the layer, without opening the settings each time, there is a "Legend" element, which is added through the menu of the elements.

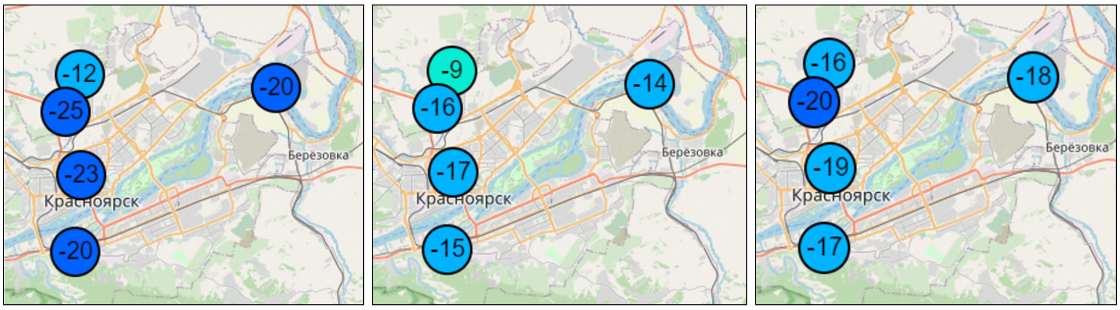


Figure 6: Thematic map with temperature data at monitoring posts for March 1, 2 and 3.

References

[1] M. Boulos, J. Warren, J. Gong, P. Yue, Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping, International Journal of Health Geographics 9 (2010). doi:10.1186/1476-072X-9-14.

[2] M. Vahidnia, H. Vahidi, Open community-based crowdsourcing geoportal for earth observation products: A model design and prototype implementation, ISPRS International Journal of Geo-Information 10 (2021). doi:10.3390/ijgi10010024.

[3] S. Yamashkin, E. Yamashkina, A. Yamashkin, Integration of knowledge, visualization and dissemination of spatio-temporal data through geoportal systems, in: 2021 Ural Environmental Science Forum on Sustainable Development of Industrial Region, UESF 2021, Chelyabinsk, volume 258, 2021, p. 03001. doi:10.1051/e3sconf/202125803001.

[4] O. Yakubailik, A. Kadochnikov, A. Tokarev, WEB Geographic Information System and the Hardware and Software Ensuring Rapid Assessment of Air Pollution, Optoelectronics, Instrumentation and Data Processing 54 (2018) 243–249. doi:10.3103/S8756699018030056.

[5] J. Jornet-Monteverde, J. Galiana-Merino, J. Soler-Llorens, Design and implementation of a wireless sensor network for seismic monitoring of buildings, Sensors 21 (2021). doi:10.3390/s21113875.

[6] S. Wiemann, J. Brauner, P. Karrasch, D. Henzen, L. Bernard, Design and prototype of an interoperable online air quality information system, Environmental Modelling and Software 79 (2016) 354–366. doi:10.1016/j.envsoft.2015.10.028.

[7] C. Brendel, R. Dymond, M. Aguilar, An interactive web app for retrieval, visualization, and analysis of hydrologic and meteorological time series data, Environmental Modelling and Software 117 (2019) 14–28. doi:10.1016/j.envsoft.2019.03.003.

[8] K. Gobakis, A. Mavrigiannaki, K. Kalaitzakis, D.-D. Kolokotsa, Design and development of a Web based GIS platform for zero energy settlements monitoring, in: 9th International Conference on Sustainability and Energy in Buildings, SEB 2017, Chania, Crete, volume 134, 2017, pp. 48–60. doi:10.1016/j.egypro.2017.09.598.

[9] A. Knörchen, G. Ketzler, C. Schneider, Implementation of a near-real time cross-border web-mapping platform on airborne particulate matter (PM) concentration with open-source software, Computers and Geosciences 74 (2015) 13–26. doi:10.1016/j.cageo.2014.10.003.

[10] F. Mao, K. Richards, M. Toland, Y. Shi, D. Hannah, S. Krause, Rivervis: A tool for visualising river ecosystems, Computers and Geosciences 123 (2019) 59–64. doi:10.1016/j.cageo.2018.11.007.