University of Tehran Press

Journal of the Earth and Space Physics

Homepage: https://jesphys.ut.ac.ir

Vol. 48, No. 4, Winter 2023, P. 151-160
(Research Article)

Internet Application for Interactive Visualization of Geophysical and Space Data: Approach, Architecture, Technologies

Vorobev, A.V.[1] ✉ | Soloviev, A.A.[2] | Pilipenko, V.A.[3] | Vorobeva, G.R.[4]

1. **Corresponding Author**, The Geophysical Center of the Russian Academy of Sciences, Moscow, Russia. E-mail: geomagnet@list.ru
2. The Geophysical Center of the Russian Academy of Sciences, Moscow, Russia. E-mail: a.soloviev@gcras.ru
3. Schmidt Institute of Physics of the Earth of the Russian Academy of Sciences, Moscow, Russia. E-mail: pilipenko_va@mail.ru
4. Department of Geophysical Research Methods (Department of Geophysics), Faculty of Mining and Oil, Ufa State Petroleum Technological University, Ufa, Russia. E-mail: gulnara.vorobeva@gmail.com

(Received: 5 Nov 2022, Revised: 30 Nov 2022, Accepted: 20 Feb 2023, Published online: 5 March 2023)

Abstract

The proposed software solution is a tool developed for the analysis, forecast and visualization of geophysical data, which is collected and provided by a set of spatially distributed heterogeneous data repositories via standard web protocols (HTTP, HTTPs, FTP, etc.). They include ground magnetic observatories and stations, satellites, as well as various numerical models based on geophysical standards and specifications. The technological stack is limited with the tool's web-based implementation and represented by integrated client- and server-side technologies with specialized frameworks and APIs. Client-side implementation is represented by several markup, styling and interaction software technologies, which are HTML5/CSS3/JavaScript with geospatial ESRI ArcGIS API for JavaScript available as the RESTful resources. Django web framework based on the "Model – View – Controller" architectural model represents server-side implementation, where Python is the main programming language used for the application's business logic. The complete Web-based GIS represents a web portal with a set of services providing a rich instrumentation for the appropriate geophysical data analysis, processing, and visualization. Each tool upon execution provides an interactive geospatial image, which is generated according to the user request parameters or by default date-time settings. The proposed web services are freely available at https://aurora-forecast.ru and https://geomagnetic.ru through the web browsers.

Keywords: Rich internet applications, Geoinformation technologies, Geophysical data, Geomagnetism, Space weather.

1. Introduction

The continuously growing volume and complexity of geophysical data is an urgent challenge to the existing programming tools for the data processing, analysis, and visualization (Engebretson et al., 2019; Dawson et al., 2013; Soloviev et al., 2013, Soloviev, 2018; Kudin et al., 2021). Under the current conditions these tools do not effectively cope with the solution of scientific and practical tasks assigned to them. Large amounts of data, the need to maintain their consistency and integrity, the importance of ensuring a multi-user mode of the software operation, together with the need to ensure its high processing speed are the factors that determine the need to radically revise existing approaches.

Currently, the NOAA web service (https://www.swpc.noaa.gov/products/aurora-30-minute-forecast) is the most widespread for short-term forecasting of the intensity of aurora and visualization of the probability of atmospheric glow in the region of auroral ovals. The service is based on the statistical model OVATION-prime (Newell et al., 2014).

Also a number of less popular web services are known that are primarily focused on regional monitoring of fragments of the auroral oval. They include software developed by the University of Alaska Fairbanks, USA (https://www.gi.alaska.edu/monitors/aurora-forecast) and the Icelandic Meteorological

Cite this article: Vorobev, A.V., Soloviev, A.A., Pilipenko, V.A., & Vorobeva, G.R. (2023). Internet Application for Interactive Visualization of Geophysical and Space Data: Approach, Architecture, Technologies. *Journal of the Earth and Space Physics*, 48(4), 151-160. DOI: http//doi.org/10.22059/jesphys.2023.350281.1007467

Publisher: University of Tehran Press.
DOI: http//doi.org/10.22059/jesphys.2023.350281.1007467

Print ISSN: 2538-371X
Online ISSN: 2538-3906

Service (https://en.vedur.is/weather/forecasts/aurora/)
Analysis of the existing software solutions revealed several disadvantages. First, when receiving static spatial images, there is no support for real time interaction with the user. As a result, data consumers are faced with the impossibility of dynamic image scaling and using auxiliary spatial layers. Second, there are no tools for spatial analysis of the visualized parameters (e.g., reverse geocoding, measuring distances and areas, etc.). Since the visualization is limited by the forecast horizon (~ 30 min), there is no possibility of analyzing retrospective data.
Thus, it is necessary to develop a publicly available web-oriented service that provides an interactive user interface for visualizing predictive and retrospective geophysical data, as well as their prompt and retrospective analysis to increase the efficiency of monitoring and forecasting the considered geophysical phenomenon. Apparently, the modern geographic information technologies and related programming environments and libraries provide a powerful toolkit for the required solutions assuming further development and scaling.

2. Initial data

The considered software solutions operate with spatial data sets only. Each spatial data object is defined by two components (Kolios et al., 2017). Its spatial component consists of a pair of geodetic angular coordinates, and its attributive component contains descriptive information in a numerical table format. The latter might include both static (e.g., geodetic height of an object) and time dependent (e.g., geomagnetic field variations) quantities, thus enabling combined spatiotemporal data analysis. The presence of temporal characteristics is required in numerous geophysical studies including retrospective processing, analysis and visualization of historical data.
Initial data are often heterogeneous, distributed among numerous data providers and in fact, "black boxes" being incompatible in format with the proposed software solutions and derived web applications. In this case, the original spatial data are accessed using one of the well-known data transmission protocols on the Web. Typically, HTTP/HTTPS or FTP protocols provide output data in CSV-like or JSON format (Kim, 2020) for subsequent accumulation, processing, and analysis on the server side. This enables accessing the source data via a URL that is passed as an input parameter to the method aimed at retrieving data from remote sources. Here remote sources are understood as sources that are physically located on servers different from the one where the developed web application that uses them is running.

3. Functionality

The developed software performs spatiotemporal processing, analysis and visualization of geophysical data in several steps. First, it forms a list of both local and remote data sources, defines their formats and auxiliary characteristics (metadata, sampling rate, etc.). Next, it generates an integrated information flow consisting of heterogeneous data flows from different data sources. The obtained spatiotemporal data array goes through the next processing stage, which includes normalization, reconstruction and other options defined by a user.
The client side of the proposed web application enables different ways of spatial data visualization. Those include rendering of a set of contour lines (isolines) overlaid upon selected geographic base layer. Building isolines is a complicated and computationally expensive process. Thus, its implementation on the client side should be avoided due to its negative impact on the application processing speed. Therefore, the isolines are composed on the server side from the integrated data array at the stage of data processing. To improve visualization, the spatial data are interpolated in advance so that the isolines are derived from interpolated data. Upon completion of this stage, the control is further transferred to the client side of the application.
The client component aims at providing a user with the convenient interface for analyzing and visualizing spatial data with an adjustable time reference. At the same time, the implemented web environment makes this functionality available without being tied to a specific user workstation. When the application is loaded, various data visualization options

become available in a web browser window with default values. For example, the current timestamp is set as the default value for time parameter. As a user changes any of the spatial, temporal or attribute parameters using interface elements, a request is sent to the server and web rendering is reconfigured.

The web rendering of spatiotemporal data received from the server is based on the corresponding virtual cartographic object. The applied options include a three-dimensional virtual globe or a two-dimensional flat map. The data transmitted from the server are rendered on such an object as a set of scalable and switchable spatial layers. It enables us to apply the capabilities of GIS technologies for their control and interaction with attribute information of the spatial items (Kolios et al., 2017). Also, the application involves direct and reverse geocoding functions, which make it easier for users to reach the desired point or area at the Earth's surface upon specifying either geodetic coordinates or a full-text address, which is further converted by the API into the corresponding spatial features.

The data that are not georeferenced but have a time reference (including multidimensional time series) are visualized using interactive graphs placed on the application page and adapted to user-specified or default query parameters.

4. General architecture

The proposed Web-GIS tool is based on a traditional three-tier client-server architecture. The client-side scripting implements the geospatial data visualization as well as a functionality for a user interaction with the web application. The server-side tier of the Web-GIS is responsible for data collection, processing, and analysis. The requested data set is packed into the response and sent to the client side via a network connection.

The interaction between the client-side and server-side tiers is carried out via the HTTP (s) protocol using the exchange of messages. Basic cartographic data are transmitted via HTTP protocol from a WMS (Web Map Service) server, while the rendered data is presented in JSON (JavaScript Object Notation) format (Kim, 2020).

The general workflow of the system is described as follows. Throughout the lifecycle of an application, back-end processes on the server side serve for interacting with distributed data providers. According to the specified protocol, with a 5-minute interval, connection sessions are established with the providers' services. In the course of these sessions, the data are requested, processed and placed in the cloud storage. We use PaaS-type cloud services as it provides flexible scaling and full-featured support from the corresponding API (Application Programming Interface).

Further, on the server side, the regular grid is processed by converting into a JSON array of the "coordinates - attribute values" format for a given timestamp. Data processing is based on the output of the cloud storage request in accordance with the time parameters that were received in the body of the user request. The generated array of spatial coordinates and attribute values is fed to the input of the module for a regular grid processing. Here, the data interpolation and the construction of a set of contours are performed in accordance with the Delaunay triangulation algorithm (Isaaks & Mohan, 1989). Contours in the form of a JSON data stream are transmitted as a server response to the client side of the application (Figure 1).

One of the issues to be solved in the course of the Web-GIS implementation is the choice of technology stack. The principal requirements to the programming technologies of the application implementation include:

- Possibility to process, analyze and visualize geospatial data;
- Client-server implementation;
- High processing speed of large data amounts enabling web application ergonomics.

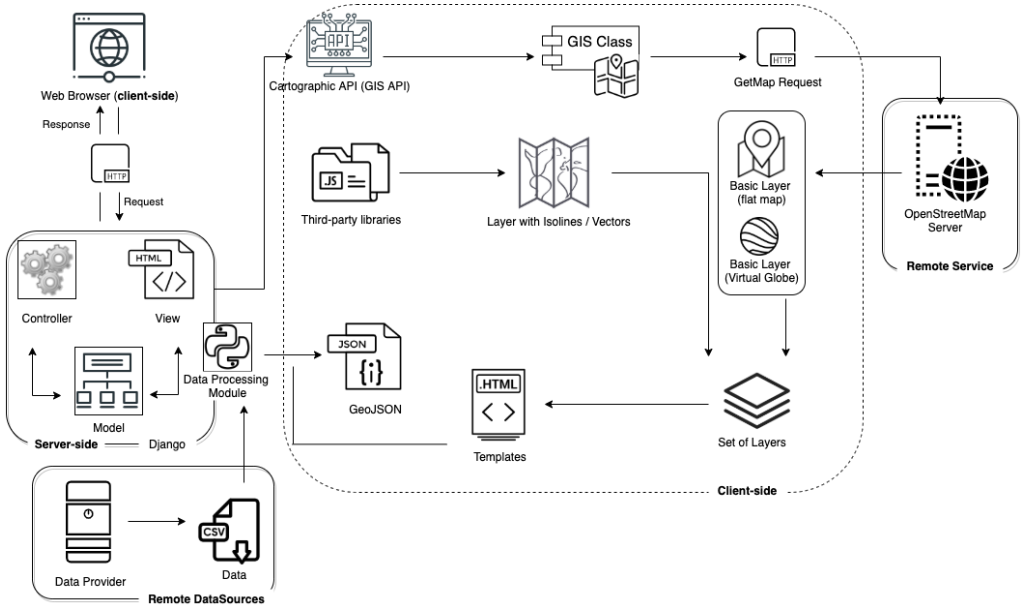


Figure 1. General architecture of the proposed software.

5. Client-side architecture

We analyzed various APIs and their programming libraries aimed processing and visualizing geospatial data of different types and formats in order to choose an appropriate client-side scripting technology for implementation of the required functionality. As a result, we selected 12 APIs potentially suitable for the desired web-implementation (Table 1), although only one of them satisfied all the defined requirements. We fixed upon ArcGIS API for JavaScript, which is a lightweight API for embedding maps, geospatial analysis and visualization features into a web application. It includes CDN-support, Dojo-kind JavaScript classes, multiple modes for the map visualization (two-dimensional as a flat map and three-dimensional as a virtual globe), wide range of applications and hardware accelerations (Liu & Yan, 2021; Bhatia et al., 2019). It also provides a wide range of geostatistical methods and algorithms that disburden a developer of programming sophisticated mathematical and statistical operations for large amounts of geospatial data.

The client-side implementation based on the ArcGIS API for JavaScript implies embedding of the appropriate programming classes into parametrical templates, which are used at the presentation layer of the three-tier architecture based on the MVC-pattern. Each geospatial iteration is implemented via the programming class with a set of special attributes and methods. For example, a virtual globe (Nebiker et al., 2010) represents an instance of the "SceneView" class, which is designed for 3D cartographical imaging using WebGL. Here WebGL stands for Web Graphics Library, and its primary feature is rendering two- and three-dimensional web content without using any other plugins or libraries. An instance of "SceneView" class inherits the properties and methods for geospatial data rendering in a web application, as well as the methods for necessary geostatistical computations (mapping projections, coordinate convertation, etc.). The geospatial visualization using the "SceneView" class is based on Document Object Model (DOM) of the web page, which provides uniquely identified web interface object as a place for the virtual globe rendering. The appropriate ID is sent as an input parameter to the constructor of ArcGIS "SceneView" class. According to the proposed scheme, each spatial object assumes generation of the DOM-object instance. However, for security reasons, modern web browsers limit available DOM-object instances (about 500 instances for most web browsers), whereas each DOM-object has its own virtual copy in the virtual memory. In the project, we generate a pair of geospatial objects with the one for the virtual globe visualization, and another for the layer with the rendered geospatial data. The latter results from the processing and geospatial analysis of the source geophysical data.

In addition to spatial data visualization, the

client-side scripting also supports multifunctional toolkit for user interaction with the data. It includes instruments for inverse and reverse geocoding, which enables searching for a location according to the parameters of natural-language request, as well as retrieving full-text object description based on specified geographical coordinates. The application also includes geolocation option, which provides a user with the proper geographical coordinates of his device location. The data are retrieved as both geographical latitude and longitude format and full-text description in standard GIS format "zip-code, country, city, street, building". Other interface elements provide basic functionality including zooming options for controlling the visible fragment of the virtual globe or the flat map.

The application enables visualizing the solar terminator line between the Earth's daylight and dark night sides, which is rendered on either three-dimensional virtual globe or two-dimensional flat map. This feature is implemented using the appropriate class of ArcGIS API for JavaScript. The terminator position depends on the UTC timestamp, which is defined by default or by a user.

To make the application interface user-friendly at different devices the client-side markup is programmed adaptively by combining HTML5 and CSS3 elements within Bootstrap framework. The framework is included in the project using CDN-technology, so it does not take any additional disc space to be implemented in the web application. The markup implementation is based on the so-called "grid" principle, which adapts the user interface elements to the appropriate device screen size and resolution without any loss of the content, graphical design and functionality implemented on the client side (JavaScript-scripts, various APIs, etc.). In addition, some ready-to-use interface elements are inherited from the Bootstrap framework including switches, date and time pickers, and others.

Client-side interactive elements also include charts, which graphically represent numerical data as time series retrieved from the server-side scripts of the application. In the course of the development, we obtained a key to use Google Charts API, so the further programming process was fully CDN-based. Again, it saves of allocating extra physical space on the server for running the application. Also, this kind of approach increases the processing speed of the application, as the client-side scripting is limited to caching only static content elements. Those include several functions, classes and methods in APIs, which are changed rarely. Every chart is generated by a separate function based on input time series.

Table 1. Modern GIS-APIs.

Name (URL)	Programming Languages Support			Visualization Mode	Free to Use	Web Access	Hardware Acceleration
	JavaScript	Python	Java				
ArcGIS API (https://developers.arcgis.com/)	+	+	+	Map, Globe	+*	+	+
Bing Maps V8 Web Control (https://www.microsoft.com/en-us/maps/web/)	+	–	–	Map	+*	+	–
Cesium (https://cesium.com/)	+	–	–	Map, Globe	+	+	+
Gio.js (https://giojs.org/)	+	–	–	Globe	+	+	+
Google Maps Platform (https://developers.google.com/maps/apis-by-platform)	+	–	–	Map, Globe	+*	+	–
Leaflet (https://leafletjs.com/reference-1.7.1.html)	+	–	–	Map	+	+	–
NASA World Wind (https://worldwind.arc.nasa.gov/)	+	–	+	Globe	+	+	+
OpenGlobus API (https://www.openglobus.org/)	+	–	–	Globe	+	+	+
OpenLayers (https://openlayers.org/en/latest/apidoc/)	+	–	–	Map	+	+	–
WebGLEarth (https://www.webglearth.com/)	+	–	–	Globe	+	+	+
WhirlyGlobe (https://mousebird.github.io/WhirlyGlobe/)	+	–	–	Map, Globe	+	+	+
Yandex Maps (https://yandex.ru/dev/maps/)	+	–	–	Map	+*	+	–

Note: * shareware (with restrictions on the number of requests per day).

6. Server-side architecture

The server-side implementation is totally Python-based. This choice is conditioned by the wide range of geospatial data operations and the need for their high processing speed. The Python script is available for both desktop applications and web applications on the server side within the Django framework (Damyanov, 2019; Potapov et al., 2019). The latter is a popular server-side framework for the rich internet applications development. Particularly, it supports architectural template (and a programming template also) named MVC (Model–View–Controller). The MVC-template allows the developers to separate the business logic from the data and the user interface of the web application. This type of the architecture makes the web application more open for applying the elements of microservice architecture, as all the necessary external programming scripts can be included into the project (interface, business logic, data storage and processing, etc.).

The developed web application aimed primarily at geophysical data processing utilizes several external services. They include an URI-based query for the spatiotemporal data in the auroral zone of the Earth for both retrospective and prognostic analysis. The query supports program-to-program and program-to-human operational modes, which makes this function available in the external applications and for the simple user's queries in the client agent (e.g. web browser) without using specific programming or interface tools. Another considered external service enables analyzing geophysical parameters derived from the Weimer empirical model, which provides high-latitude electric and magnetic fields, the associated currents and Joule heating as a function of the solar wind parameters (Weimer, 2005). This URI-based service is a POSIX-like query with a set of spatial and temporal input parameters, and its output result represents a text stream in the JSON-format.

The programming services are implemented as stand-alone modules fully on the server side of the application. Although they are physically located and interpreted within the developed application, they can still be used from the outside. This provides a clear separation of the application business logic from its interface (i.e. client) side, which is one of the significant advantages of the developed programming architecture. At the same time, such approach does not support so-called reactive applications (a separate type of applications managed by e.g. React.js), so this feature is to be implemented in the future.

Here we should mention the minor role of the database (model) in the implemented architecture of the web application. Although the MVC template of the web application assumes using the database, its traditional interpretation as a relational or object-relational data model is not applied here. The reason is that the project is based on the distributed system of heterogeneous data sources, which cannot be represented by the same data model. The computational models used to derive data are neither logical nor physical databases but implemented as functions based on a set of input spatial and temporal parameters.

Data visualization process is implemented on both client and server sides of the application. For spatial data interpolation, a dedicated Python-based script is running on the server side. The script gets the input data according to the user (or other client) request parameters from the data stream or computational models. Next, the data are algorithmically interpolated using Delaunay triangulation with a specified precision value and the spatial layer is generated as a stream of geoJSON data. On the next step, the geoJSON-stream is sent to the client side. The client-side script based on the ArcGIS API for JavaScript interprets the stream of spatial data and executes processing and visualization algorithms using the user-defined model (for example, based on three-dimensional virtual globe or as a flat map).

Several advantages of the data representation in geoJSON format for its processing, analyzing, and visualizing on both client and server sides should be mentioned. geoJSON is an extension of the widely used and fully supported JSON format. It is an open format for representing and storing geospatial data as a set of attributive information and geospatial features. The text-based geoJSON format uses natural language (English) elements, utilizes a very simple data structure, and is easy for both humans and

machines to read. The ArcGIS API for JavaScript supports this format for graphical representation of geospatial data as a layer (or a set of layers) with necessary interactive tools.

Delaunay triangulation used for spatial data interpolation is useful for defining the location of a point by measuring angles from known points at either end of a fixed baseline. In this project, it is used to determine the attribute value at the point based on a set of the known values from the neighboring points. Common Delaunay triangulation assumes that for X points there is a triangulation T(X) such that for any triangle T(X) there are no points from X within the bounds of its circumcircle. Notably, if X points are in general position, then a unique Delaunay triangulation for X exists. The class method from the Matplotlib.tri library is used for building an unstructured triangular mesh from the geographic coordinates of the original spatial data. As an output, the method retrieves an array of values corresponding to a triangular grid. Additionally, a rectangular regular coordinate grid is formed from the original data coordinates using the Meshgrid method of the Numpy class.

Due to uneven resolution of original data, their interpolation often leads to noises in the derived set of isolines, which are visually represented as peculiar creases on the graphic layer. For their elimination and noise reduction, we apply a Gaussian filter (Young & Van Vliet, 1995). To eliminate the stepping effect in the displayed isolines, a smoothing filter is applied. Again, we apply the "Gaussian Filter" class of the SciPy library. The input of this class is a set of interpolated attribute data Z' and the output is a filtered data set Z''.

7. Quality Assurance

We performed the application quality assessment based on the indicators of its reliability, maintenance, ease of use, efficiency, versatility and correctness. The testing of the application was carried out under normal and extreme conditions as well as exceptional situations. According to the results of computational experiments, the application performs its functions correctly. In extreme and exceptional situations, the application displays messages with the corresponding errors and continues running as usual. In addition, the response time of the software system was estimated by comparing with the widely used method, based on the monolith architecture and synchronous single-point rendering (Rhyne & MacEachren, 2004) for client and spatial data processing.

The experiments were carried out using a personal computer with an average performance characteristic: 1.6 GHz processor with two cores, 4 GB RAM, Internet connection speed 20 Mbps. The web server configuration was 72 * Intel (R) Xeon (R) Gold 6140 CPU @ 2.30GHz. According to the experimental results, the execution of a data query and their rendering into the isolines takes on average about 5.5 s when using the proposed solutions versus about 49 s when using the widely accepted method of synchronous single-point rendering [16], which provides 88.8% time savings.

8. Applications

The proposed software solutions are widely used to solve several practical problems related to the processing, analysis and visualization of spatial geophysical data with a given sampling rate obtained from heterogeneous sources (Figure 2).

The first application was developed to visualize the distribution of the probability of observing the aurora and the electric field potential of the polar ionosphere in the northern and southern hemispheres of the Earth. The web application was developed at the Geophysical Center of the Russian Academy of Sciences and is available at https://aurora-forecast.ru/ (Figure 2, a, b) (Vorobev et al., 2020a; Vorobev et al., 2020b).

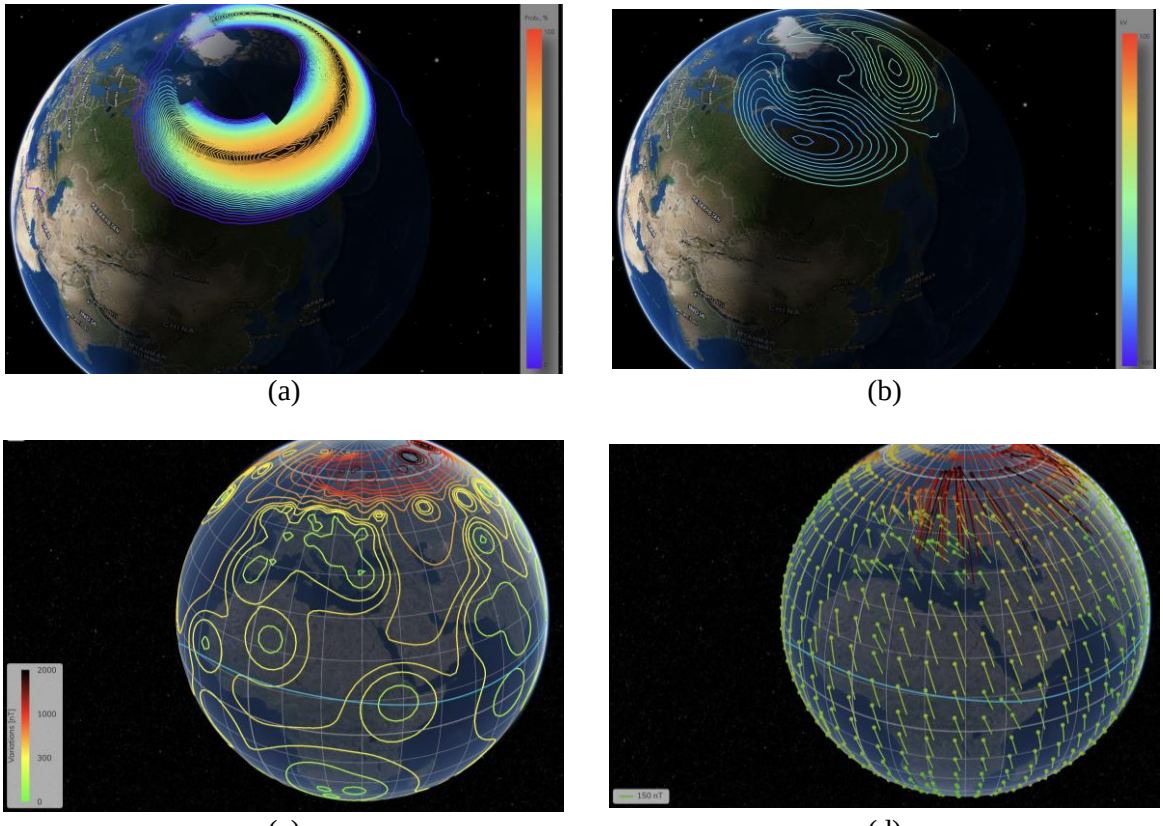
The forecast of the geophysical parameters of the auroral oval is based on mathematical models (Newell et al., 2009; Newell et al., 2010; Newell et al., 2014) parametrized by the solar wind and interplanetary magnetic field parameters measured in real time by ACE (https://services.swpc.noaa.gov/json/ace/) and DSCOVR (since 2016) (https://services.swpc.noaa.gov/json/dscovr/) satellites. These empirical models include

OVATION (Oval Variation, Assessment, Tracking, Intensity, and Online Nowcasting) model for the short-term forecast of the global integrated radiation of energy (Newell et al., 2014) and Weimer model (Weimer, 2005). Here we apply them to calculate the electric and magnetic potential in the auroral zone and determine the boundaries of the auroral oval.

The application visualizes on a virtual globe a layer of spatial isolines characterizing the auroral oval location and probability of observing auroras in accordance with the time parameters specified by the user. A distinctive feature here is the ability to visualize chronologically data preceding online information by pre-filtering data already available on the server. A complex of geoinformation tools including the reverse geocoding function, dynamic scaling and measuring tools provides user's interaction with the application.

Another web application based on the proposed software solutions provides processing, analysis and visualization of the spatial distribution of geomagnetic field disturbances and its variability. The developed web application is available at https://geomagnetic.ru/ and is currently open for beta testing. The initial data are provided by the SuperMAG project in the geomagnetic coordinate system N (geomagnetic North), E (geomagnetic East), Z (vertical). Spatial data interpolation is carried out using the Inverse Distance Weighting (IDW) method (Vorobev et al., 2020c).

The application includes three services that provide visualization of geomagnetic field variations in time, space and dynamic mode (Figure 2, c, d) (Vorobev et al., 2020c). Having initially a small number of data sources, the implemented methods are reliable enough to identify the regions most prone to extreme geomagnetic disturbances. In the Arctic region, where space weather manifestations are the most significant, these regions are of particular interest as geomagnetic extreme events can cause harm to various types of technological infrastructure facilities.



(a) (b)

(c) (d)

Figure 2. Examples of the proposed Web-GIS: a, b) Distribution of the probability of observing the aurora and the electric field potential of the polar ionosphere as of 2021/03/20 12:00UT; c, d) Distribution of the disturbed total intensity (isolines) and the horizontal component (vector field) of the geomagnetic field as of 2015/03/17 23:45UT.

9. Conclusion

One of the crucial tasks of processing, analysis and visualization of spatiotemporal geophysical data is the improvement of the existing software and the underlying architecture in terms of the efficiency, processing speed and ergonomics. Combining modern geoinformation and web development technologies contributes significantly to this. The paper presents a new approach to the implementation of the software architecture of a web-based system for processing, analyzing and visualizing spatiotemporal information by the example of geophysical data. The approach is based on a combination of the traditional client-server three-tier architecture and the innovative microservice architecture for developing web applications. At the same time, the majority of calculations for retrieving data from remote sources, their integration and processing are performed on the server side. Each of these features is a microservice available for using by third-party web services and applications. In addition, on the server side, within the framework of the proposed architecture, the functionality of the third-party software modules is utilized, which fits into the concept of a microservice architecture.

On the client side, the proposed software architecture enables visualization of spatiotemporal information. The available visualization options include flat two-dimensional map and three-dimensional virtual globe. The data generated on the server side of the application is transmitted to the client side and displayed on a given graphic cartographic object (flat map or globe) as a separate layer, which may include a set of spatial isolines (contour lines).

To assess the efficiency of the proposed software solutions, two web applications were deployed and are now available in the beta-testing mode. The first one is designed to visualize the probability of observing auroras at the Earth's surface (https://aurora-forecast.ru). The second one provides processing, analysis and visualization of the spatial distribution of geomagnetic field disturbances and its variability (https://geomagnetic.ru). In the course of the computational experiments using typical web server and personal computer, the processing speed of the web applications was estimated by comparing with the performance of the widely accepted solutions. The analysis of the assessment results has revealed an increase in the processing speed of the web applications by an average of 9 times, which suggests the feasibility of applying the proposed approach for a wider range of geophysical applications.

Acknowledgments

The facilities of the GC RAS Common Use Center "Analytical Center of Geomagnetic Data" (http://ckp.gcras.ru) were used for conducting the research. This study was supported by the grant No. 21-77-30010 from the Russian Science Foundation.

References

Bajaj, D., Bharti, U., Goel, A., & Gupta S.C. (2009). PaaS providers and their offerings, *International Journal of Scientific & Technology Research*, 9(2), 4009–4015.

Bhatia, T., Singh, H., Litoria, P., & Pateriya, B. (2019). Web GIS Development using Portal for ArcGIS. ArcGIS Server and Web AppBuilder for ArcGIS, 10, 43-47.

Damyanov, I. (2019). Data Aggregation in Microservice Architecture. *International Journal of Online and Biomedical Engineering (iJOE)*, 15. 81.

Dawson, E., Lowndes, J., & Reddy, P. (2013). The British Geological Survey's New Geomagnetic Data Web Service. *Data Science Journal*, 12, WDS75-WDS80.

Engebretson, M. J., Pilipenko, V. A., Ahmed, L. Y., Posch, J. L., Steinmetz, E. S., Moldwin, M. B., Connors, M. G., Weygand, J. M., Mann, I. R., Boteler, D. H., Russell, C. T., & Vorobev, A. V. (2019). Nighttime magnetic perturbation events observed in Arctic Canada: 1. Survey and statistical analysis, *Journal of Geophysical Research: Space Physics*, 124(9), 7442-7458.

Isaaks, E.H., & Mohan R. (1989). An Introduction to applied geostatistics – Oxford: Oxford University Press, 592 p.

Kim, J.R. (2020), A Study on the JSON Compatible Serialization Standard Method of LPG Data, *The Journal of Next-Generation Convergence*

Technology Association, 4(6), 581–588.

Kolios, S., Vorobev, A., Vorobeva, G., & Stylios, Ch. (2017). GIS and environmental monitoring. *Applications in the marine, atmospheric and geomagnetic fields. Cham, Switzerland: Springer International Publishing AG*, 2017, 174 p.

Kudin, D. V., Soloviev, A. A., Sidorov, R. V., Starostenko, V. I., Sumaruk, Yu. P., & & Legostaeva O. V. (2021). Advanced production of quasi-definitive magnetic observatory data of the INTERMAGNET standard, *Geomagnetism and Aeronomy*, 61(1), 54–67.

Liu, Z., & Yan, T. (2021). Comparison of spatial interpolation methods based on ArcGIS. *Journal of Physics: Conference Series*, 012050.

Nebiker, S., Bleisch, S., & Gülch, E. (2010). State of the art and critical issues Virtual Globes. *GIM International*, 24(7), 17–21.

Newell, P. T., Liou, K., Zhang, Y., Sotirelis, T., Paxton, L. J., & Mitchell, E. J. (2014). OVATION Prime-2013: Extension of auroral precipitation model to higher disturbance levels. *Space Weather*, 12, 368–379.

Newell, P. T., Sotirelis T., & Wing S. (2009). Diffuse, monoenergetic, and broadband aurora: The global precipitation budget. *J. Geophys. Res.*, 216, A09207.

Newell, P. T., Sotirelis, T., & Wing S. (2010). Seasonal variations in diffuse, monoenergetic, and broadband aurora. *J. Geophys. Res.*, 115, A03216.

Potapov, V. I., Shafeeva, O. P., Gritsay, A. S., Makarov, V. V., Kuznetsova, O. P., & Kondratukova, L. K. (2019). Reliability in the model of an information system with client-server architecture. *Journal of Physics: Conference Series*, 1260, 022007.

Rhyne, T.-M., & MacEachren, A. (2004). Visualizing geospatial data, 31.

Soloviev, A., Khokhlov, A., Jalkovsky, E., Berezko, A., Lebedev, A., Kharin, E., Shestopalov, I., Mandea, M., Kuznetsov, V., Bondar, T., Mabie, J., Nisilevich, M., Nechitailenko, V., Rybkina, A., Pyatygina, O., & Shibaeva, A. (2013). The Atlas of the Earth's Magnetic Field (Eds.: A. Gvishiani, A. Frolov, V. Lapshin), Publ. GC RAS, Moscow, 361 pp.

Soloviev, A.A. (2018). Methods of geoinformatics and fuzzy mathematics in geophysical data analysis, *Chebyshevskii Sbornik*, 19(4), 194-214, (in Russian).

Vorobev, A.V., Pilipenko, V.A., Enikeev, T.A., & Vorobeva, G.R. (2020a). Geoinformation system for analyzing the dynamics of extreme geomagnetic disturbances from observations of ground stations. *Computer Optics*, 44(5), 782-790.

Vorobev, A. V., Pilipenko, V. A., Krasnoperov, R. I., Vorobeva, G. R., & Lorentzen, D. A. (2020b). Short-term forecast of the auroral oval position on the basis of the “virtual globe” technology. Russian Journal of Earth Sciences, 20(6), ES6001.

Vorobev, A.V., Pilipenko, V.A., Reshetnikov, A.G., Vorobeva, G.R., & Belov, M.D. (2020c). Web-oriented visualization of auroral oval geophysical parameters. *Scientific Visualization*, 12(3), 108–118.

Weimer, D. R. (2005). Improved ionospheric electrodynamic models and application to calculating Joule heating rates. *Journal of geophysical research*, 110, A05306.

Young, I., & Van Vliet, L. (1995). Recursive implementation of the Gaussian filter. *Signal Processing*, 44(2), 139-151.