Scientific Visualization, 2024, volume 16, number 2, pages 81 - 96, DOI: 10.26583/sv.16.2.07

Interactive Geoinformation System for Dynamic Visualization of Auroral Oval Characteristics Based on Component-Oriented Programming Patterns

A.V. Vorobev[1,A,B], G.R. Vorobeva[2,B]

[A] Geophysical Center of the RAS, Moscow, Russia
[B] Ufa University of Science and Technology, Ufa, Russia

[1] ORCID: 0000-0002-9680-5609, geomagnet@list.ru
[2] ORCID: 0000-0001-7878-9724, gulnara.vorobeva@gmail.com

Abstract

The practical need for operational monitoring of the position dynamics of the auroral oval is due to the negative technospheric manifestations of space weather effects observed in this spatial region. However, the well-known software tools for modeling the auroral oval solve this problem inefficiently from the point of view of informativeness and ergonomics. The paper proposes an approach to dynamic visualization of the characteristics of the auroral oval, which is available at the program level in the form of a traditional web application with the ability to render visual elements in the browser, as well as in the format of a standalone service like RESTful-API. The developed solution provides visualization of the following parameters: the probability of observing the glow of the upper layers of the atmosphere with the naked eye, the electric and magnetic potentials of the field in the region of the northern auroral belt, as well as various types of auroral precipitation. It is assumed that the proposed approach will make it possible to significantly increase the efficiency of the study of parameters in the auroral oval region by specialists and scientists in the relevant fields. At the same time, the high internal and low external connectivity of the developed software modules allow them to be integrated into third-party applications of various profiles and purposes.

Keywords: auroras, auroral oval, geoinformation systems, component-oriented approach, API, web application.

1. Introduction

Auroras are a clear confirmation of the invasion of charged particle streams into the atmosphere and direct evidence of the existence of solar-terrestrial connections. Their comprehensive research is currently determined by the urgent needs of practice.

It is known that auroras visually appear at night mainly in the northern and southern polar caps, and in some cases - to the equator from the aurora zones. According to known standards (for example, standard 25645.109 84 [1]), auroras are most often observed in the area of the auroral oval - an area of the ionosphere that is a projection of the plasma layer and cusp along the geomagnetic field lines. In this case, it is important to distinguish between the so-called instantaneous and averaged auroral ovals, the first of which characterizes a ring actually observed above the Earth’s surface, and the second - a certain oval figure, the size and shape of which is determined by the configuration of the magnetosphere and the parameters of the solar wind [2].

Research shows that during magnetically quiet periods the diameter of the auroral oval is ~3000 km, while on the day and night sides the boundary of the auroral zone is 10-16° and 20-23° from the magnetic pole, respectively. During periods of solar activity, the auroral oval expands and auroras can be observed 20-25° south or north (for the northern or southern hemisphere, respectively) of the boundaries of their usual manifestation.

The auroral oval region, as a rule, is characterized by the most active manifestations of space weather, which include, in particular, overload of power lines caused by geo-induced currents, failures of short-wave radio communication systems, etc. [3]. In addition, auroral latitudes are characterized by the presence of sharp gradients and high levels of ionospheric plasma turbulence, which provokes failures and reduces the stability of signals from radio communication systems and GPS/GLONASS navigation satellite systems [4].

In this regard, the importance of monitoring of the dynamics of the auroral oval position for assessing associated phenomena is obvious. Moreover, from the point of view of the possibilities of predicting the position of the auroral oval, the assessment of the corresponding probability can be carried out with finite accuracy in space and time. The very possibility of the forecast is due to a finite time shift (~1 hour) due to the propagation of the solar wind from the interplanetary satellite to the magnetosphere boundary, information about which is the basis for modeling the position of the auroral oval.

The main source of information about the structure of the auroral oval is measurements on low-orbit satellites of auroral electron fluxes that cause auroras. The results of such measurements do not depend on ionospheric illumination and atmospheric cloudiness, and are also more sensitive compared to ground-based and satellite optical observations. Thus, in particular, the OVATION-prime (OP) auroral oval model, based on data from more than 20 years of observations of electron and proton fluxes of different energies on DMSP (Defense Meteorological Satellite Program) satellites, has become widespread [5].

Automated monitoring of the position of the auroral oval based on well-known empirical models is implemented by a number of currently existing information systems, which are primarily web-based services. So, for example, the NOAA web service (https://www.swpc.noaa.gov/products/aurora-30-minute-forecast) [5], which is based on the OVATION-prime model, has become widespread and is used by the National Oceanic and Atmospheric Administration (NOAA) [5] for short-term forecasting of the intensity of auroras and provides visualization of the probability of airglow in the area of the auroral oval (Figure 1, a). The visualization results provided by the service are actively used both by scientific organizations in the process of conducting various types of research, and by travel agencies to attract tourists to the high-latitude regions of the planet to observe the auroras.

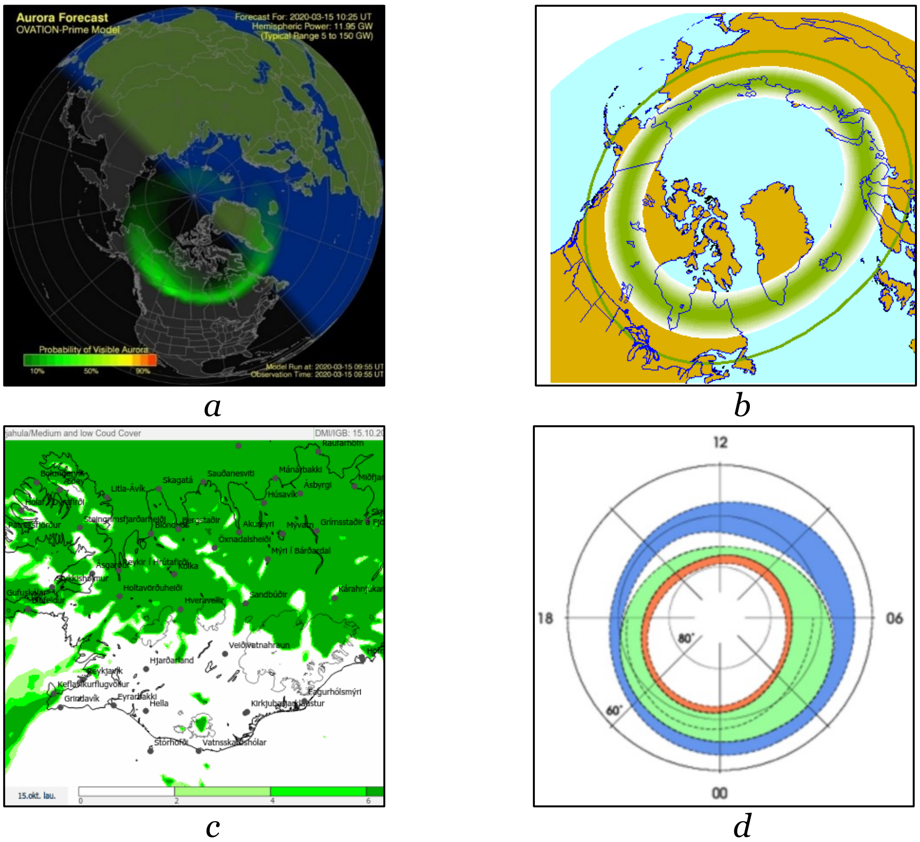


Figure 1 – Examples of visualization results for auroral oval models

In addition to the above-mentioned NOAA service, there are other software products that are focused primarily on regional monitoring of fragments of the auroral oval. A great example of such web applications are products developed by the University of Alaska (Fairbanks, USA) (https://www.gi.alaska.edu/monitors/aurora-forecast) (Figure 1, b), as well as the Icelandic Meteorological Service (https ://en.vedur.is/weather/forecasts/aurora/) (Figure 1, c).

Another Russian software product is built on the basis of the Auroral Precipitation Model (APM) [6, 7] and is available at URL: http://apm.pgia.ru/. The model is built on a database of geomagnetic activity, represented by Dst and AL index values, and allows one to obtain the global distribution of the characteristics of precipitating electrons in the coordinates "corrected geomagnetic latitude - local geomagnetic time" (Figure 1, d). It is important to clarify here that the Dst index represents the deviation of the magnetic field variation from the quiet level, averaged over the values measured at the control chain of magnetic stations located at low latitudes, and the AL index corresponds to the maximum negative deviation of the H component of the magnetic field from the average quiet level at stations of the auroral zone. From the point of view of software implementation, this service does not allow building the specified model for real current values of Dst and AL-indices (they are supposed to be set manually), and is also characterized by the lack of interactivity of images, cartographic substrates and support for geoinformation tools, which together reduces the effectiveness of the tool.

In addition, the NORUSCA model developed within the framework of the Norwegian-Russian project for forecasting the characteristics of the auroral oval using data from the virtual 15-minute Kp index (WING), which, in turn, is determined based on the dependence of the Kp index on solar wind parameters, is known [8, 9]. It is also important to clarify here that the Kp index is a planetary index that characterizes the global disturbance of the Earth's magnetic field in a three-hour time interval. The model allows to construct an auroras oval for 1 to 2 hours in advance, depending on the speed of movement of charged particles from the Sun. The corresponding results are available at http://kho.unis.no. The application is a single-user type and is available only in desktop or mobile format.

Summarizing the experience of research of the above and other similar software products, we can highlight a number of their characteristic shortcomings. These include, in particular, the impossibility of dynamic visualization and scaling, the lack of visual information about the current state of space weather, as well as geographic information tools for manipulating analysis parameters and graphical interpretation of spatial data.

In this regard, it is relevant to develop an interactive geographic information system that provides dynamic visualization of the parameters of the auroral oval with the possibility of their user analysis using geoinformation methods and tools based on real values of space weather parameters. It is expected that the solution to the identified scientific and technical problem will make it possible to develop tools for a better understanding by researchers and interested parties of the physics of various types of processes in auroral and adjacent zones.

The experience gained by the authors in the field of software development [10, 11] led to the conclusion that the basis for solving this problem should be the use of geoinformation and web-based technologies, which will expand the circle of application users, on the one hand, and will also significantly reduce the requirements for client computing power, on the other.

2. Characteristics of input data

The project discussed in the paper is aimed at visualizing the following parameters in the region of the auroral oval: the probability of observing the glow of the upper layers of the atmosphere with the naked eye, the electric and magnetic field potentials in the region of the northern auroral belt, as well as various types of auroral precipitation. Computational models that generate the corresponding spatial data sets are implemented in the form of executable software scripts located on the server side of a web-based application.

The initial data are the results of satellite observations, available via standard network data transfer protocols and provided by providers in accordance with a given time stamp in the form of a set of attribute and spatial information. The data provided by the services is formatted in text data streams in CSV-like and/or JSON format [12], which allows them to be efficiently processed in appropriate server scripts.

So, for example, one of the visualized parameters which is the probability of observing auroras with the naked eye can be determined in accordance with the OVATION-prime predictive model, which in turn implements a short-term (30 min) forecast of auroras based on space weather and solar wind parameters. In this case, a delay value of 30 minutes corresponds to a solar wind speed of ~800 km/s; however, in reality, the delay time varies from less than 30 minutes to an hour or more, depending on the average solar wind speed.

The model itself is implemented at the level of software scripts executed on the server side and returns a text dataset in GeoJSON format as a result [12]. Each geospatial primitive (spatial point) in the corresponding server response represents a combination of three components - geographic latitude and longitude (spatial data) and the probability value of observing the aurora at the corresponding point in space. Data is transmitted via the secure HTTPS protocol via port 443 in a standard request-response format [13].

Another parameter—the boundaries of auroral precipitation—is formed based on the results of a server script that implements the APM auroral precipitation model [6, 7], which, in turn, was obtained from direct observations of the characteristics of precipitation particles from DMSP series satellites. The input information for the scenario is the values of the magnetic activity indices AL and Dst, which can be obtained from the resource of the World Geomagnetic Data Center in Kyoto (WDC for Geomagnetism, Kyoto (http://wdc.kugi.kyoto-u.ac.jp/wdc/ Sec3.html)) [14]. A significant drawback of this resource is the lack of data from February 28, 2018, which does not allow constructing a planetary distribution of various auroral precipitation zones in later time periods. In the research prototype of the application presented by the authors, it is proposed to use the values of AL and Dst indices of magnetic activity according to forecast data from the Laboratory for Atmospheric and Space Physics at the University of Colorado (USA), available at: https://lasp.colorado.edu/home/spaceweather/ [15].

The data is presented in the format of a CSV text file consisting of two fields: a timestamp of the form “YYYY/DD-HH:MM:SS” (DD is the serial number of the day in the year) and the directly predicted value of the corresponding magnetic activity index. Due to the limited availability of the designated resource via web protocols for third-party applications, in the project under consideration, access to data is implemented through additional software extractors of text information using the corresponding APIs.

The parameters of the electric and magnetic potential in the auroral zone are calculated based on the Weimer model [16, 17], which is also implemented in software as a server script module, initiated for execution by a request from the client side of the corresponding application. The input data for the specified software scenario are the parameters of the solar wind and interplanetary magnetic field recorded by the DSCVR satellite (https://www.ngdc.noaa.gov/dscovr/portal/index.html). It is important to note that, in accordance with Weimer's model, spherical harmonic functions can be used to determine potential values only in a narrow region of high latitudes. At lower latitudes, potentials are calculated from several functions of longitude in the Fourier series with a discrete step in latitude [18]. The corresponding data is presented in a file independent of the program script itself, which is accessed via the same secure HTTPS network protocol via port 443.

3. Visualizing Modelling Results

In general, the results of modeling parameters in the auroral oval region are sets of spatial data with corresponding attribute values. Visualization and processing of such data is traditionally carried out using geographic information systems and technologies that provide the ability to use a wide range of tools, models and methods of geoinformatics.

The basis of a geospatial image is a basic geospatial object with a corresponding map background. As a rule, such objects are flat two-dimensional maps or virtual three-dimensional globes. Since this project focuses on the auroral oval region, it seems appropriate to use three-dimensional geospatial visualization. This choice is due to the fact that virtual globes using the Mercator map projection (WGS 84) have the best technical characteristics for visualizing spatial data in the upper and lower latitudes compared to flat maps, where such visualization is in most cases accompanied by graphic artifacts [19].

Directly visualized data on a virtual globe should be represented by a set of geospatial graphic primitives, which in most cases include a spatial point, a broken line (polyline) and a polygon. Due to the specificity of the analyzed data in the area of the auroral oval, in the project under consideration, visualization is implemented using geospatial broken lines and polygons. At the same time, each variant of geospatial interpretation has its own color scheme, which makes it possible to increase the efficiency of visual perception of the displayed information by the end user.

The choice of the type of geospatial primitive for visualization is determined by the specifics of the interpreted data [20]. So, for example, to visualize the probability of observing the glow of the upper layers of the atmosphere with the naked eye, as well as the electric and magnetic field potential in the region of the northern auroral belt, the project under consideration uses a set of broken lines. In this case, polylines are level lines that are formed from initial spatial data, represented by spatial points with corresponding attribute values. To generate level lines (isolines), a separate software module is provided on the server side of the application, which takes as input a set of spatial points and values in them, and returns a set of isolines as a result. The specified script implements the corresponding spatial interpolation algorithms for irregular monitoring networks, and also directly performs the procedure for generating level lines from spatial data and filtering the result obtained.

Another geospatial primitive used in the project is a spatial polygon, which is used to visualize various types of auroral precipitation. The result of executing the server script that implements the APM model consists of three spatial polygons, each of which corresponds to a zone of auroral precipitation (area of hard diffuse precipitation, auroral oval, area of soft diffuse precipitation). The color scheme is applied to the generated spatial polygons directly on the client side of the application.

The generation and visualization of sets of graphic primitives on the client side are implemented in a dynamic mode, which allows one to evaluate changes in time and space of the corresponding parameters in the area of the auroral oval. This option provides for a time-sequential (with a user-specified sampling step) change of spatial layers characterized by the corresponding spatial label. The absence of layer overlay during the visualization procedure creates for the end user the effect of animated switching of geospatial images, and the ability to stop the procedure allows you to study the data for the corresponding time interval in more detail.

The settings available in the application allow to switch the color scheme of the data visualized using geospatial primitives in the auroral oval region. To make the geospatial image more informative, a visualization of the terminator linked to the corresponding time stamp is provided. In this case, if necessary, to increase the readability of the geospatial image, the terminator can be disabled.

In accordance with the Web 2.0 concept [21], control of the visualization of geospatial layers in the auroral oval region is implemented in asynchronous mode, which is reflected in the corresponding format of interaction with the server: the object model tree of a web page with a virtual globe does not change completely, but only partially, affecting hide the layer without reloading the geospatial base object with the map underlay.

To simplify user navigation in the visualization mode of the analyzed parameters, the application provides mechanisms that implement geoinformation methods of direct and reverse geocoding, which allows you to shift the focus of the image to the desired spatial point. In addition, with appropriate settings of user agents, a geolocation function is possible, allowing you to move the focus directly to the geographic location of the user device.

4. Application Architecture

The web-based framework of the application ensures its mass distribution and accessibility to a wide range of end users with minimal qualifications in the field of information technology. At the same time, this imposes a number of serious restrictions on technologies for software implementation of the functions of processing, analysis and visualization of relevant geospatial information. This fact is appropriately reflected in the architecture of the presented web application.

The traditional approach to designing web applications is based on the use of a three-tier client-server architecture, in which a web server is used as a mediator component, which respectively manages the interaction between the client and server components. At the same time, the client-server architecture pattern successfully implements the concept of separating data from its presentation, providing for the possibility of redistributing computationally complex functions between several server nodes with the possibility of subsequent vertical and/or horizontal scaling to increase the reactivity of the resulting applications [22].

It is known that in web applications, increasing reactivity, testability, flexibility and extensibility is only possible by weakening the relationships between software modules on both the client and server sides. Consistent hierarchical decomposition is the basis of the component-oriented approach that is widespread today, providing the ability to reuse autonomous (or loosely coupled) software components, including third-party applications [23].

A distinctive feature of the component approach to application development is the development and operation of software components that are autonomous relative to each other within a given environment. In this case, the environment is understood as the computing environment in which the web application operates: platforms, frameworks, code interpreters and compilers, etc. However, the architecture itself is designed in such a way that its components do not depend on each other and are easily replaceable during refactoring and/or scaling the application.

Each component of the application has characteristic properties, the most important of which is introspection, which presupposes that each component has metadata necessary and sufficient for its use as part of the application when interacting with other software modules. As a rule, the metadata is the component interface, which regulates the set of input parameters, as well as a set of output data both in the context of the domain used and syntactic features.

To implement the necessary functionality in the application under consideration, components are identified that ensure the formation of a set of spatial data, on the one hand, as well as components that implement their cartographic interpretation, on the other. At the same time, in accordance with the principles of modular decomposition of web applications, groups of components implement client and server components, respectively, as well as objects that ensure interaction with external data sources.

Processing and graphical interpretation of geospatial data is usually associated with computationally expensive procedures. In accordance with the plugin approach widely practiced in modern web applications, it is advisable to identify a group of server components responsible for preparing and processing spatial data. At the same time, the performance and reactivity of server software components is directly determined by the degree of their external and internal connectivity.

It is known that it is possible to optimize the functioning of a web application if, for example, the principle of openness/closedness is implemented: "Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification." This is intended to ensure flexibility and extensibility of the software system, which, on the one hand, implies the ability to quickly make changes to the program, as well as the ability to add new entities and functions to the system without disturbing its basic structure. At the same time, this architecture also provides the ability to reuse software modules in third-party systems.

When decomposing a software system, it is necessary to strive for minimal dependence of modules on each other, on the one hand, and maximum dependence of the internal components of a module (High Cohesion + Low Coupling principle) [24]. At the same time, the high interconnectivity of the components within the module is manifested in the fact that the module is focused on solving one narrow problem and does not imply the implementation of heterogeneous functions.

A wide range of component-oriented architectural approaches are actively practiced in modern development, which are mainly reflected in the use of specialized design patterns for web applications. The most widespread among the patterns that provide strong internal and weak external coherence of software components are the "Observer" and "Mediator" architectural patterns [25].

Thus, the "Observer" pattern assumes the presence in the application architecture of an additional control software module, conventionally designated as an "observer module". The role of the observer module in the application is reduced to the formation of a generalized message coming from internal or third-party components of the architecture, and intended for the corresponding local program modules. Interaction between software modules is implemented according to the "sender-subscriber" principle, according to which the main module is the sender of messages, and all other modules are its subscribers. The message metadata indicates the identification parameters of the desired destination module (usually a URI). In the background, modules monitor the appearance of new messages and, if the metadata matches their own identifier, they accept the corresponding packet.

Another template – "Mediator" – also assumes the presence of an additional control module (the so-called mediator). However, unlike the previous approach, the messages generated during interaction are not publicly available and are sent by the mediator directly to the recipient module, thereby relieving all other software components of the need to continuously monitor all changes in incoming information flows.

Both architectural patterns assume the presence of an additional software module (observer or mediator) that manages information interaction between other application components. The key advantage of this approach is ensuring maximum independence of application components, which allows them to be distributed and used according to the web API principle in accordance with the RESTful software interaction method (REpresentational State Transfer) [26].

So, for example, a set of spatial isolines generated on the server side, characterizing the probability of observing auroras in a regular spatial grid, has a characteristic API interface. This method of interaction allows you to vary the corresponding data visualization by various (including those other than local in relation to the virtual directory) software services and systems using the standard secure HTTPS protocol.

Moreover, each software component developed within the framework of the proposed solution encapsulates the features of the computational processes embedded in it in such a way that a set of corresponding boundary parameters was sufficient for its identification. So, for example, the formation of a resulting set of spatial polygons in accordance with the APM model is possible using a software component, the boundary values of which are a timestamp, Dst and AL index values, on the one hand, and the resulting GeoJSON data, on the other.

From the point of view of web-oriented architecture, the proposed solution is an application based on the "model-view-controller" pattern. The pattern implements a key design principle of separating data from presentation. At the same time, an ORM (Object-Relational Mapping) model is provided between data coming from third-party sources and business logic, which provides for the unambiguous mapping of information into the model based on the corresponding data schema.

Control actions within the framework of the proposed scheme (Fig. 2) are formed from the side of the controller module and are initiated by user actions or requests from third-party / local software modules. In this case, the model (according to the "Observer" pattern) is in continuous interaction with the data through periodic and trigger messages, responses to which initiate the sending of corresponding messages to the controller.

It is important to note that in the proposed scheme the controller acts as a mediator that interacts with other software modules. At the same time, the software components present in the architecture with their metadata are registered in the module registry associated with the mediator and provide there, in particular, the URI address necessary for direct access.

Thus, in accordance with the proposed architecture, a web-based application is a collection of software components with weak external connections and strong internal coherence. Each component can be used independently from other software modules, receiving control from internal or third-party software components according to the provided input parameters.

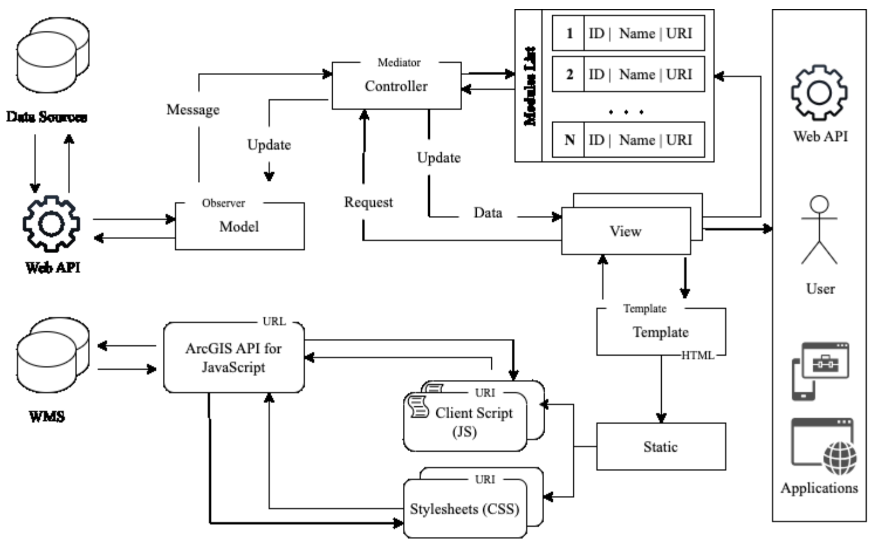


Figure 2 – Web GIS Architecture

Each of the software components in the proposed solution is assigned its own unique URI, which is also a parameter of its absolute addressing. This parameter provides support for the GET method for setting input parameters, which allows you to access the corresponding software component using a standard network protocol.

In general, there is a RESTful interpretation of each software module, which assumes its web API architecture. In this case, access to the module is implemented (and encapsulated) through a standard software interface, the descriptors of which must be transmitted as component metadata in the header of the corresponding message.

5. Program implementation of the solution

To qualitatively and quantitatively evaluate the effectiveness, as well as test the proposed solution, a research prototype of a web-based geographic information system was developed, providing visualization and geostatistical analysis of geophysical parameters in the area of the auroral oval. The presented prototype is in the public domain and is accessible at the URL https://aurora-forecast.ru using the standard secure HTTPS protocol via a web browser installed on the user's computer.

The stack of software technologies used is represented by the Django framework, which provides the use of the Python programming language for developing server scripts, on the one hand, and coding the client component of the application based on the traditional HTML5 / CSS3 / JavaScript combination, on the other. The framework by default implements the Model-View-Controller (MVC) architectural pattern, thereby ensuring the implementation of the principle of separating data from its presentation.

In addition to the MVC design pattern, the proposed approach of combining the "Observer" and "Mediator" patterns was introduced into the architecture of the developed web GIS, which as a result made it possible at the software level to achieve autonomy of the designed software modules, respectively ensuring the implementation of various processing, analysis and visualization operations geospatial information. At the same time, the designated software modules, taking into account their implementation, ensure use as RESTful services, which is possible due to the availability of universal access to them through a unique URI.

Program modules registered in the architecture are represented in the urls.py component, which, in accordance with the principles of the MVC structure of the Django project, is classified as a registry of modules that the controller directly operates (in this case, in the role of a mediator / intermediary). Acting as an intermediary, the controller module ensures the relative independence of program modules from each other, transmitting the corresponding information flows according to the metadata of the module registry.

In addition to the standard HTML/CSS/JavaScript client bundle, which implements user interface elements for organizing the interaction of the application consumer with rendered spatial data, a number of third-party software libraries are connected to the developed web GIS. At the same time, due to the relative stability of the corresponding content, CDN (content delivery network) technology is used for this purpose, expanding the project in question to third-party libraries in their original source. This approach, on the one hand, is characterized by high performance without using its own computing power, and ensures continuous updating of software mechanisms, on the other.

One of the most important third-party components of the developed web GIS is the ArcGIS software library, which is accessible through a web API and provides the developer with a powerful tool for geospatial analysis and visualization. The specified library is intended for use in JavaScript scripts on the client side and is available via a CDN connection.

6. Interface of the Solution

The developed web GIS is built on the principle of designing SPA applications (Single Page Application) and is a dynamic web page visually divided into two functional areas. One of them is responsible for displaying the parameters of the solar wind and space weather in the form of interactive graphs, where the time axis indicates the values of time stamps in UTC format, and the y-axis represents the corresponding values of the analyzed parameters.

The second component of the developed interface is a direct cartographic image of geophysical parameters in the area of the auroral oval. The central element of this component is an interactive virtual globe with a basic cartographic background loaded from a remote WMS server.

On a virtual globe in the format of dynamically loaded layers, geophysical parameters selected by the user are visualized. Each dynamic layer option has its own color scheme, which is described on the same visual panel in the form of a so-called cartographic legend.

To ensure the user's interactive work with the application, support for a number of geoinformation tools is implemented. For example, in the developed application, direct and reverse geocoding functions are available, which allows the end user to quickly access a geospatial point using the full-text name of the corresponding geoposition. In addition, Web GIS supports fast geospatial visualization tools and displays the cursor position on the screen converted to geodetic coordinates (latitude and longitude) to the user.

Parameters for displaying a spatial layer on a virtual globe can be set by the user through the control panel, where options are available for switching between layers, as well as controlling the color scheme (monochrome or complementary palette), terminator, etc. To avoid blocking the application while calling the server for spatial data, all requests are implemented in asynchronous mode.

Another important function of the developed application is the ability to retrospectively forecast relevant geophysical data. To do this, the presented web GIS provides a control element that allows you to select the desired date and time in UTC format. Geospatial rerendering allows the user to visually analyze the spatial distribution of data given a given timestamp.

The developed application provides the user with the ability to visualize the spatial layer, including in retrospective mode. To do this, you need to select the desired date and time in UTC format using the visual element presented in the application interface. In this case, directly in the date selection element itself, at the program level, days are deactivated for which it is impossible, for one reason or another, to generate a set of spatial data.

Another distinctive feature of the application is support for dynamic visualization of spatial layers with analyzed parameters. For this purpose, the interface provides an element of the "TimeSlider" class, which allows the user to automatically and/or manually switch between layers corresponding to certain timestamps. When working with the application, the user selects the starting and ending time points for analysis, as well as the sampling step (at this stage, a value from 5 to 30 minutes can be selected).

The developed application provides dynamic visualization of spatial layers according to two parameters - the probability of observing the glow of the upper layers of the atmosphere with the naked eye, as well as the electric and magnetic field potentials in the region of the northern auroral belt. The user-selected visualization option (monitoring network parameter), which by default acts as the base for creating an integrated layer, is considered relevant for calculating time dynamics.

It should be noted that executing a request for dynamic visualization involves sending a series of requests to third-party sources using an established protocol. Each received response leads to the formation of a set of spatial data corresponding to one point in time. Upon completion of a series of requests to third-party (or local, depending on the parameter), a single integrated spatial layer with corresponding time parameters is formed.

Switching spatial frames using the TimeSlider class control is focused on working with the generated integrated spatial layer. The software script associated with the specified control parses the spatial data of the integrated layer and divides it into frames in accordance with the specified time interval. The tools available to the user for continuous or discrete viewing of spatial frames provide the possibility of appropriate dynamic visualization.

At the software level, the dynamic rendering function is directly linked to controlling the visibility of spatial layers, which allows for a smooth transition between rendering frames, each of which corresponds to a specific timestamp. It is important to note that when implementing this function, a single integrated data layer is formed, and each graphic primitive in it is assigned its own time parameter value. The redundancy characteristic of this approach is a necessary necessity and has virtually no effect on the reactivity of the application as a whole.

The component-oriented architecture of the developed application allows it to be used in two versions. First, end users can access the application through a browser and interact with it through client-rendered controls. At the same time, interaction with the server in asynchronous mode makes it possible to ensure fairly high ergonomic characteristics of the application. On the other hand, the component-oriented approach provides a functional decomposition of the system into separate autonomous software modules. Each of them, in accordance with the principles of the specified architecture, is given its own unique URL. The latter, in turn, can be used by third-party software systems and provides connection to third-party projects (not only with a web interface) of corresponding software modules.

At the same time, it seems appropriate to note that in the project, at the level of each software module (in this case we are talking mainly about the server side of the application architecture), CDN (Content Delivery Network) support is deliberately disabled. This allows you to avoid possible collisions associated with caching of the corresponding software services for similar client requests to them.

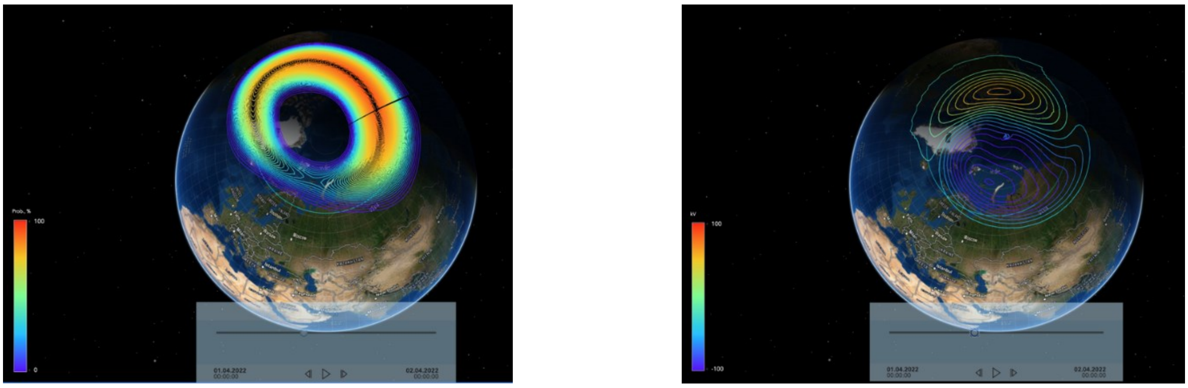
Each autonomous software module is focused on generating a specific set of spatial data based on given input parameters, which are primarily date and time values. The result of executing the software module is a spatial layer in GeoJSON format, available both for visualization by well-known geographic information systems and technologies, and for analysis using specialized geostatistical libraries.

7. The discussion of the results

The web-based GIS developed in accordance with the proposed solutions is hosted and freely available to users at https://aurora-forecast.ru. To work with the application, the user needs a browser and a stable connection to the Internet.

Support for client-side hardware acceleration of web graphics enables rendering of high-quality spatial images with varying levels of detail. The main spatial object in this case is a three-dimensional virtual globe with support for interactivity and the ability to scale.

Figure 3 shows screen forms of the application using the example of dynamic visualization of thematic spatial layers. Thus, Figure 3, a presents a variant of visualizing the probability of observing auroras in the form of a dynamic layer. The formation of the layer is implemented in accordance with user-specified time stamps of the beginning and end of the analyzed period. The result of the server request is a spatial layer decomposed into frames, each component of which corresponds to a specific time stamp from a user-specified range. A visual element at the bottom of the user screen allows you to control the display of spatial frames in continuous or discrete mode.



a *b*

Figure 3 – Screen forms for dynamic visualization of spatial layers

A similar example is shown in Figure 3, b, which presents a variant of visualization of the electric and magnetic potential in the auroral zone in the northern hemisphere. The rendering is carried out by analogy with the previous example and provides the ability to visually analyze a dynamic spatial image with various scaling options and frame-by-frame playback modes.

The static visualization mode involves the formation of a geospatial image in accordance with a user-specified time stamp. Depending on the selected parameter for visualization, the corresponding layer is formed in the form of spatial isolines. Management of the generated geospatial image is implemented through appropriate tools, including mechanisms for direct and reverse geocoding, geostatistical analysis, color scheme management, etc. (Fig. 4).

To assess the quality of the developed geographic information system, a series of tests were carried out, aimed both at identifying functional inconsistencies in the operation of the application and at analyzing its behavior in various situations. Experimental studies were carried out on the client side using a computer (CPU Intel Core i5 10300H GHz, RAM 4 GB, Internet connection speed ~52.4 Mbit/s) and on the server side - based on a web server with a 72 * Intel(R) processor Xeon(R) Gold 6140 CPU @ 2.30 GHz.

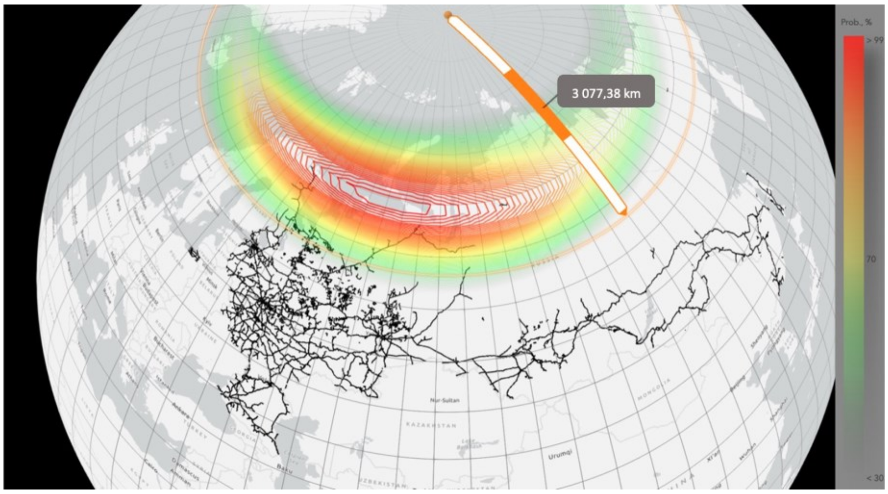


Figure 4 – Screen form of static visualization of spatial layers

The performance and fault tolerance of the application were assessed through load testing. The results of the studies showed that the maximum performance of the software system on the presented configuration was 15 connections (4,816 processed requests) / hour.

The quality assessment of the application in terms of compliance with actual and expected functionality was carried out in accordance with standard 28195-89. In accordance with this methodology, the reliability, maintainability, ease of use, efficiency, versatility and correctness of the software system were assessed. Testing was carried out under normal and extreme (situations related to the lack of input data were considered) conditions, as well as exceptional situations (low-speed Internet connection, high load on the server, manually entering request parameters for the server). According to the results of computational experiments, the software system performs its functions correctly. In extreme and exceptional situations, the program displays appropriate error messages and continues to operate normally.

8. Conclusion

The relevance of creating an interactive information system for visualizing geophysical parameters in auroral latitudes is largely determined by the need to monitor the position of the auroral oval in the decision-making process in applied areas. It is precisely these areas that are characterized by the most pronounced manifestations of space weather, destructive for systems and objects of the technosphere (for example, failures in radio communication systems and GPS/GLONASS navigation satellite systems).

Analysis of well-known software products showed their low efficiency due to the impossibility of scaling and dynamic visualization, lack of interactivity and geoinformation profile tools. The problem is aggravated by the lack of aggregated data on the current state of space weather on known resources.

In this regard, a web-based interactive geoinformation system has been proposed and developed, providing dynamic visualization of the parameters of the auroral oval with the possibility of their user analysis using geoinformation methods and tools based on real values of space weather parameters.

From the point of view of software implementation, a distinctive feature of the proposed solution is an architecture based on a component-oriented approach. The low external and strong internal coherence of program modules achieved in this way makes it possible to use them as independent services (based on the RESTful API principle), on the one hand, and increases the extensibility of the application as a whole, on the other.

From a functionality point of view, the developed application provides visualization of the following parameters in the auroral oval region: the probability of observing the glow of the upper atmosphere with the naked eye, the electric and magnetic field potentials in the northern auroral belt, as well as various types of auroral precipitation. Visualization of spatial layers is available in static and dynamic modes, which allows you to both analyze the values of the corresponding parameters at a given point in time and evaluate their dynamics over a certain time interval with a sampling step set by the user. Support for geoinformation tools and the ability to control the appearance of a spatial image expand the capabilities of users when working with the application compared to possible analogues. Another distinctive feature is the presentation of aggregated space weather data in the application at relevant points in time, which also increases the information content of the results of the application as a whole.

In terms of quality, the results of testing the developed application showed that it correctly performs its functions. In extreme and exceptional situations, software scripts generate the necessary error messages and continue to operate as normal.

Acknowledgments

The work was supported by the Russian Science Foundation grant 21-77-30010.

References

1. Standard GOST 25645.109-84. Earth's magnetosphere. Terms and Definitions. Moscow: USSR State Committee for Standards, 1984 (in Russian)

2. Lazutin, L. L. (2015) Aurora oval – a beautiful but outdated paradigm // Solnechno-zemnaya physica, v. 1, no. 1, p. 23-35, doi: 10.12737/5673

3. Kozyreva, O. V., V. A. Pilipenko, R. I. Krasnoperov [et al.] (2020) Fine structure of substorm and geomagnetically induced currents // Annals of Geophysics, 63, no. 2, GM219.

4. Shagimuratov I., Chernouss S., Despirak I. [et al.] (2018) Occurrence of TEC fluctuations and GPS positioning errors at different longitudes during auroral disturbances // Sun and geosphere, 2018, no. 13, p. 89-94. doi:10.31401/SunGeo.2018.01.12.
5. Newell, P. T., K. Liou, Y. Zhang, et al. (2014) OVATION Prime-2013: Extension of auroral precipitation model to higher disturbance levels // Space Weather, no. 12, p. 368–379.
6. Vorobyov V.G., Yagodkina O.I. (2005) Influence of magnetic activity on the global distribution of the auroral precipitation zone // Geomagnetism and Aeronomy, vol. 45, no. 4, p. 438-444 (in Russian).
7. Vorobyov V.G., Yagodkina O.I. (2007) Dynamics of auroral precipitation during strong magnetic storms // Geomagnetism and Aeronomy, vol. 47, no. 2, p. 185-192 (in Russian).
8. Chernous S. A., Shagimuratov I. I., Filatov M. V. [et al.] (2018) Comparison of spatio-temporal distribution of TEC inhomogeneities and model auroral oval // Physics of Auroral Phenomena, vol. 41, no. 1, p. 160-163. doi: 10.25702/KSC.2588-0039.2018.41.160-163.
9. Chernous S. A., Filatov M. V., Shvets M. V. [et al.] (2018) Auroral perturbations as an indicator of ionosphere impact on navigation signals // Russian Journal of Physical Chemistry, vol. 12, no. 3, p. 562-567, doi:10.1134/S1990793118030065.
10. Vorobev A. V., Pilipenko V. A., Enikeev T. A. [et al.] (2021) System for dynamic visualization of geomagnetic disturbances according to the data of ground magnetic stations // Scientific Visualization, no. 13.1, p. 162–176, doi: 10.26583/sv.13.1.11
11. Vorobev A. V., Pilipenko V. A., Krasnoperov R. I. [et al.] (2020) Short-term forecast of the auroral oval position on the basis of the ``virtual globe'' technology // Russ. J. Earth Sci., 20, p. ES6001, doi:10.2205/2020ES000721.
12. Mooney P., Minghini M. (2022) Geospatial data exchange using binary data serialization approaches // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLVIII-4/W1-2022, p. 307-313, doi:10.5194/isprs-archives-XLVIII-4-W1-2022-307-2022.
13. Lasocha W., Badurowicz M. (2021) Comparison of WebSocket and HTTP protocol performance // Journal of Computer Sciences Institute, no. 19, p. 67-74, doi:10.35784/jcsi.2452.
14. Mandrikova O., Stepanenko A.A. (2020) Automated method for calculating the Dst-index based on the wavelet model of geomagnetic field variations // Computer Optics, 44, p. 797-808, doi:10.18287/2412-6179-CO-709.
15. Luo B., Li X., Temerin M., Liu S. (2013) Prediction of the AU, AL, and AE indices using solar wind parameters // Journal of Geophysical Research: Space Physics, no. 118, doi:10.1002/2013JA019188
16. Weimer D. R., Clauer C. R., Engebretson M. J. [et al.] (2010) Statistical maps of geomagnetic perturbations as a function of the interplanetary magnetic field // J. Geophys. Res., no. 115, p.A10320, doi:10.1029/2010JA015540.
17. Weimer D. R. (2013) An empirical model of ground-level geomagnetic perturbations // Space Weather, 11, p. 107–120, doi:10.1002/swe.20030
18. Vorobev A.V., Pilipenko V.A., Reshetnikov A.G. [et al.] (2020) Web-oriented visualization of auroral oval geophysical parameters // Scientific Visualization, no. 12.3, p.108–118, doi: 10.26583/sv.12.3.10
19. Novikova E., Palamar A., Makhonko S. [et al.] (2018) Transformation parameters between UCS-2000 and WGS-84 // Geodesy and cartography, no. 44, p. 50-54. doi:10.3846/gac.2018.1830.
20. Yassine L., Lakhrissi Y., Ali S. (2022) Geospatial Big Data Platforms: A Comprehensive Review. KN // Journal of Cartography and Geographic Information, no. 16, doi: 10.1007/s42489-022-00121-7.
21. Baytar C. (2022) Dynamics of Web 2.0: A Software Application "Web 2.0 Meter" // Gazi university journal of science, doi: 10.35378/gujs.1053604.
22. Kumar S. (2019) A review on client-server based applications and research opportunity // International Journal of Scientific Research, no. 10, p. 33857-33862, doi:10.24327/ijrsr.2019.1007.3768.

23. Kaur A., Kaur P., Kaushal P. (2020) Maintainability Procedure in Component-Based Software // Journal of Computational and Theoretical Nanoscience, no. 17, p. 5156-5161, doi:10.1166/jctn.2020.9357.

24. Tyagi B., Beg Y. (2022) An Experimental Assessment on Effects of Solid Design Principles on the quality of Software using CKJM Metric Analysis // International Journal for Research in Applied Science and Engineering Technology, no. 10, doi:10.22214/ijraset.2022.46808.

25. Bagrudeen B.A., Yuvaraj Dr. D., Shithart M. [et al] (2022). An efficient mechanism for deep web data extraction based on tree-structured web pattern matching // Wireless communications and mobile computing, p. 1-10, doi:10.1155/2022/6335201.

26. Padmanaban R., Thirumaran M., Anitha P. [et al.] Computability Evaluation of RESTful API using Primitive Recursive Function // Journal of King Saud University - Computer and Information Sciences, no. 34, doi:10.1016/j.jksuci.2018.11.014.