The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-5-2024
ISPRS TC V Mid-term Symposium “Insight to Foresight via Geospatial Technologies”, 6–8 August 2024, Manila, Philippines

# Cartographic animations, a supporting technology for education

Lev Obuhov, Evgeny Panidi

Saint Petersburg State University, St. Petersburg, Russia - panidi@ya.ru, evgeny.panidi@spbu.ru

**Keywords:** Animated Cartography, Multimedia Cartography, GIS-based Automation, QGIS.

**Abstract**

To date, basic theoretical and methodological ideas of animated cartography have been formed, as well as general schemes for animated maps compilation. However, the technological issue remains relevant in this domain due to the abundance of modern tools that can be used for compilation of animated maps. Particularly, most of currently implemented cartographic animation methodologies require the use of a variety of specialized software, including geographic information systems (GISs), vector and raster graphical editors, video effects editing software, and others. So the data processing and conversion techniques are extremely complicated in some cased, when data converted from one software platform to another.
In our study, we established investigation of popular desktop GISs functionality application as a full-cycle animated cartography environment. In such a context, we discover approaches to improve methodology of animated maps automated compilation. To prototype a full-cycle animation in GIS environment, we developed a set of animation algorithms implemented in QGIS and documented the algorithms using UML (Unified Modeling Language) diagrams.

## 1. Introduction

Modeling and displaying of objects’ and processes’ dynamics in geographical space is one of traditional issues in cartography. Currently, due to the active digitalization of society and scientific and technological progress, many different tools, technologies and opportunities have appeared and can be applied to map dynamical processes taking place in a territory. In connection to this, one of the new cartography subdomains has been developed in recent decades that is animated (or animation) cartography (Campbell, Egbert, 1990; DiBiase et al., 1992; Harrower, Fabricant, 2008). This subdomain considers general characteristics, concepts, principles, and technological schemes applied to create, design and display animated maps. Significant value of animated cartography is composed of its educational potential, as dynamical maps in many cases can make easier perception of the processes reflected in maps. Animated maps in many cases appears to be more intuitive in the meaning of map model perception (especially for not qualified in cartography users), as the moving effects being usual and perceptible as a context map visualization effect, brings additional (and extremely powerful) media flow into with-map interaction process.
The language of cartographic animations itself obeys the general laws of the map language. Static graphical variables have been developed to build symbolic systems of static maps (Bertin, 1967; Morrison, 1974; MacEachren, 2004; Berlyant, 2006) including: location, size, shape, brightness, color, orientation, texture, saturation, alignment, clarity, resolution, transparency. By analogy, dynamic graphical variables have been developed for cartographic animations (DiBiase et al., 1992; MacEachren, 2004):

1. Moment – the time when the start of the current state change is initiated
2. Duration – the real-time interval between two fixed states
3. Frequency – the number of fixed states that an object reaches in any unit of time
4. Order – structuring and organizing sequence of the moments’ displaying one after the other
5. Rate of change – the time interval of animation demonstration between two fixed states
6. Synchronization – time matching between different data sets

Since animations can be demonstrated at different speeds (in real time, accelerated or slowed down), additionally the concept of the map time scale was implemented. The time scale can be defined as the ratio of the rate of change to the duration (Berlyant, 1997; Berlyant, Ushakova, 2000; Berlyant, 2006; Harrower, 2003). According to a classification of maps by time scale into small-, medium- and fast-scale, Berlyant proposed, taking the ratio of one second to longer periods of time (hour, day, week, month, year, etc.) when implementing time scale (Berlyant, 1997).
Due to the development of technologies, practical component of animated cartography is significantly ahead of its theoretical and methodological basis that is especially true for the definition of entities and basic terms (Lisickij et al., 2014). In this way, Berlyant posed in 1997 that GIS-based mapping and the development of other technological tools have significantly expanded the possibilities of dynamics displaying using special dynamic sequences – cartographic animations (Berlyant, 1997). Later in 2000, cartographic animations still were definined as special dynamic sequences of frame maps that create a motion effect during demonstration (Berlyant, Ushakova, 2000). However currently, the definition of cartographic animations involved by Berlyant and coauthors is already outdated, since cartographic animations are created not only of basic frame maps, and in principle are not sequences of these maps in many cases. Cartographic animations have emerged as independent approach that can be defined as a mean to display changes in the characteristics of individual digital objects represented in a dynamic animated map.
Being basing on computational operations with digital objects (digital models), animated cartography, can be theoretically posed as a part of GIS-based (or geoinformation, or geoinformational) mapping (Sidorina et al., 2019), and can be

defined as a set of technologies and methods for automated creation of dynamic animated maps based on GIS databased and software environments.
In other words, animated cartography have to solve three problems matched the classic set of GIS-based mapping tasks when operating spatial (spatiotemporal) data:

1. To determine that the mapped object has changed
2. To change the object data
3. To visualize the resulting dynamics

Besides this context, there are no more or less unified methodologies developed for cartographic animations elaboration when resolving different research or production cases. Moreover as compilation of full-fledged animated maps requires specialized editing of video content and graphical effects, in most cases it assumes implementation of complex data processing chains composed of a set of software (including GISs, graphic design software, video production software, Web mapping technologies (Beresnev et al., 2022), etc.), and having wide palette of data conversion obstacles between applied software.
In-the-GIS animation generally assumed to be primitive based on the appearance and disappearance of objects. Elaboration of cartographic animations in GISs nevertheless, assumed a high degree of automation both in the meaning of geospatial data processing, and in meaning of dynamic graphical variables operating, which is the main advantage compared to the application of software complex.
Due to the high degree of automation and great flexibility of time variables management, elaboration of cartographic animations in GISs remains a promising area of research and development. Improvement of in-the-GIS animation methodologies and development of in-the-GIS animation supporting tools compose the context of our research.
When observing the animated cartography as a supporting technology for education, we have to underline that migration to the full-stack GIS-based automated animation technology can ensure wide opportunities for faster educational materials production, and making content of produced materials richer (due to the possibility for all available in a GIS data to be involved into animation production process provided in native environment). In comparison to experience gained previously using complex software stacks (Whaley, 1997) we forecast GIS-only-based animations production as less resource and time consuming and higher flexible approach.

## 2. Methods

We applied the QGIS (*https://qgis.org*) as a basic GIS software. Basing on its facilities, we identified an animated maps typology to ensure systematics in GIS-based mapping methodologies development. The typology is composed of two main groups of animations.
The first group incorporates animations that can be created using the basic data processing instruments of QGIS:

1. Object scaling
2. Object position change based upon a geometry replacement and preprocessed database of the spatial object geometries
3. Dynamic labels displaying
4. Object color and transparency change

The second group incorporates animations, which must be created using applied programing and customized algorithms:

5. Object position change based upon a movement trajectory
6. Object geometry change based upon an object state database
7. Object position change based upon a movement modelling function
8. Object geometry change based upon an geometry modeling function

Main QGIS tool responsible for the management of time variables and allowing to setup change dynamic geospatial data showing in the map is the Temporal Controller panel. The panel contains two user-set value boxes for the beginning and ending time of the animation, and the animation step (rate of change) selector. These values are stored in the QGIS environment as the special variables of the current project. In addition, the panel contains buttons needed to run and stop animation and reflects current animation state (current frame).
Another one tool directly related to dynamic spatial data visualization in QGIS is the Temporal panel in the Layer Properties window of the feature layer. This panel allows tune the layer animation mode.
Last embedded QGIS tool we applied is the Geometry Generator option for the Symbol Layer Type of the map feature layer symbology, which is an integral part of a feature layer style (Fig. 1). In the equation field of the Geometry Generator, it is possible to enter a custom value or a Python (*https://www.python.org*) function that returns a variable in a QgsGeometry geometry data format used to describe objects in QGIS, which will correspond to the specified Symbol Layer Type of the feature layer.

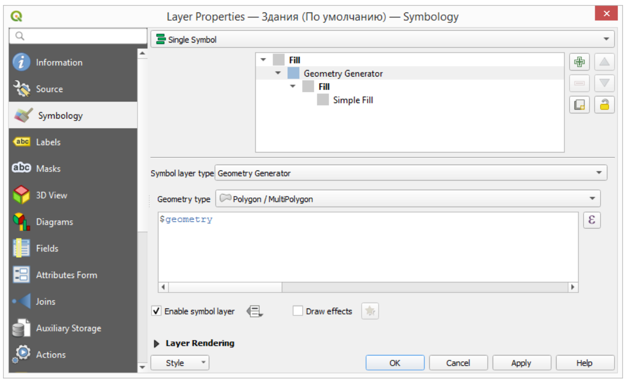

Figure 1. Geometry Generator option representation in a Symbology panel in the Layer Properties window of the feature layer.

Geometry Generator allows to create animations of object shape or position change based solely on the initial position or shape and the modeling function, with gradual drawing of intermediate shapes or positions.

## 3. Results

Basing on the typology presented in the Introduction section, we designed a series of in-the-QGIS animation methods and implemented corresponding algorithms.

The object scaling animation in QGIS is created using the built-in *scale* function, which allows to gain a new dynamic geometry of the object using Geometry Generator and applying scale conversion factor computing dependent on the animation interval. The sequence of steps to scale polygonal and linear objects (point objects are scaled by changing the size of the sign) is as follows:

1. Composing a complex style for layer objects based on the Geometry Generator
2. Compiling a custom function based on a scale transformation using the animation interval value (Fig. 2)
3. Setting the Redraw Layer Only option in the Temporal panel of the Layer Properties window
4. Setting the starting and ending time of the animation, animation step and the number of frames per second in the Temporal Controller panel

```
scale($geometry,1+minute(@map_start_time
-@animation_start_time)/100,1+minute(@
map_start_time-@animation_start_time)/
100)
```

Figure 2. Equation for the scaling function used to tune Geometry Generator.

Being tuned this way, a scaling animation can be started.

In the Geometry Generator equation, the *$geometry* parameter defines the geometry of each layer object, the scale coefficients depend on the animation interval converted to a numerical value of minutes.
For documenting purposes, we elaborated algorithm description in UML combined of class diagram describing used informational objects, variables and dependences and a block diagram describing the processing sequence (Fig. 3, 4).

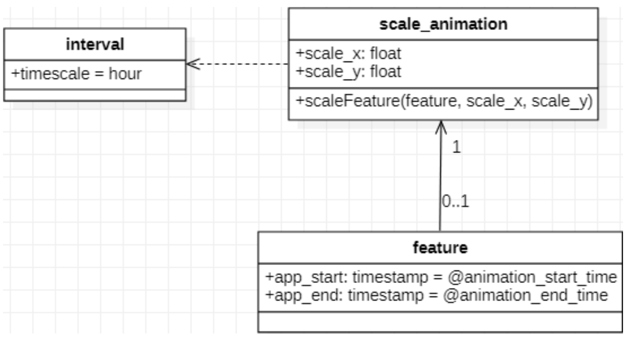


Figure 3. Scale animation UML class diagram.

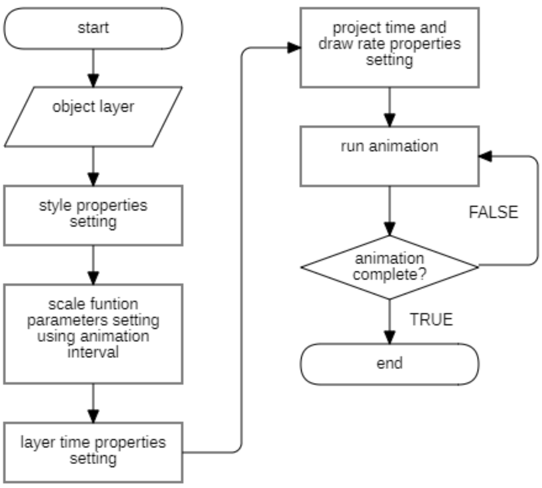


Figure 4. UML block diagram of the algorithm for creating a scaling animation.

Object position, labels displaying and object color and transparency animations were implemented in similar ways trough manipulations with built-in parameters of map object. While, the second block animations typology appears to be more complex and requires the implementation of one or more user functions (for every animation type), which assumes object geometry processing with algorithms from third-party geospatial libraries, implementing additional mathematical calculations, establishing database connections, etc.

Movement trajectory accounting when animating object position change implies the presence of a linear object describing a trajectory of motion. Every node of the object can be assigned a unique style at each moment of time, depending on geometry type of moved object (trajectory movement is applicable to all types of geometric primitives).
The movement animation along the trajectory (Fig. 5), using the positioning data of an unmanned aerial vehicle (UAV), is performed using the following list of operations:

1. Setting the Redraw Layer Only option in the Temporal panel of the Layer Properties window for a layer of linear objects used to describe trajectory
2. Creating a complex object style using Geometry Generator tuned in the Layer Properties window of a linear layer itself used to describe trajectory, for the marker used to show the UAV object position, and for the polygon object used to delineate the aerial photography coverage area (Fig. 6);
3. Creating a custom geometry processing function (Fig. 7), that returns a line with a certain number of nodes from the beginning using the Shapely geospatial data processing library (*https://github.com/shapely/shapely*), and the functions to generate the marker position and the coverage area polygon boundary
4. Embedding of the geometry processing function as a Geometry Generator equation to apply it to the three components of a complex object style (mentioned above in the item 2)
5. Setting the design of the generated objects, separately for each of the three objects (airplane marker symbol for the moved object, a solid line for the trajectory, a bordered polygon without filling for the coverage area)
6. Setting the starting and ending time of the animation, animation step and the number of frames per second in the Temporal Controller panel

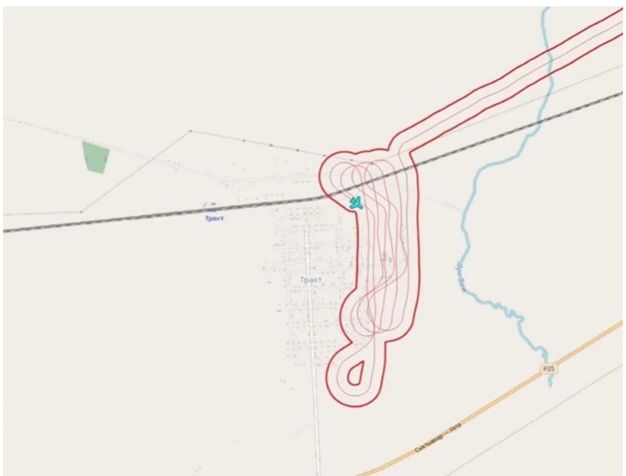

Figure 5. Visualization of along the trajectory moving animation in QGIS map.


https://doi.org/10.5194/isprs-archives-XLVIII-5-2024-69-2024 | © Author(s) 2024. CC BY 4.0 License.

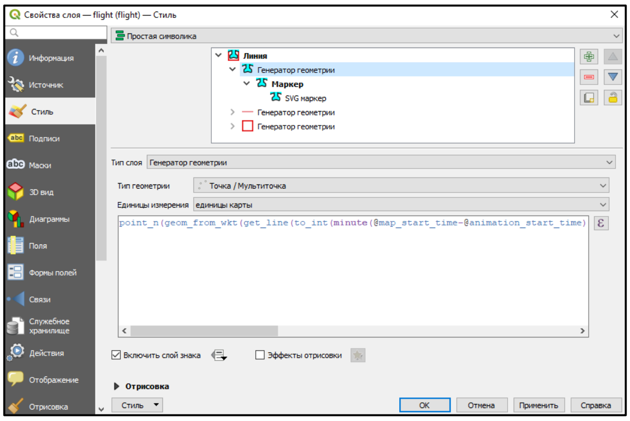


Figure 6. Geometry Generator tuned for along the trajectory movement animation.

```
@qgsfunction(args='auto',group='Custom')
def get_line(date,feature, parent):
    qgeom=feature.geometry()
    geom=wkt.loads(qgeom.asWkt())
    coors=list(geom.coords)
    if date>0:
        new_line=shapely.geometry.LineString(coors[:date])
    else:
        new_line=shapely.geometry.LineString()
    return new_line.wkt
```

Figure 7. Equation for the time dependent trajectory generation function used to tune Geometry Generator.

Geometry Generator equation (Fig. 7) is generated required geometry using built-in QGIS functions (*point_n* for the marker, *get_line* for the linear geometry of the extraction, *buffer* for the coverage area polygon generation as a certain width polygonal geometry). The equation receives as input the *date* (current animation interval, or moment), *feature* (the trajectory geometry), and *parent* (the default parameter for QGIS functions) parameters. The linear geometry is converted from the QgsGeometry format to the Shapely geometry format, which allows to process the geometry as a Python list of node coordinate pairs and limits the total number of node data in the output line. Additionally, the azimuth from the previous to the last node onto the last node of the generated trajectory is calculated to account the marker rotation when animation drawing using the built-in *azimuth* and *point_n* functions. UML class and block diagrams for the along the trajectory moving animation are presented in Fig. 8 and 9.

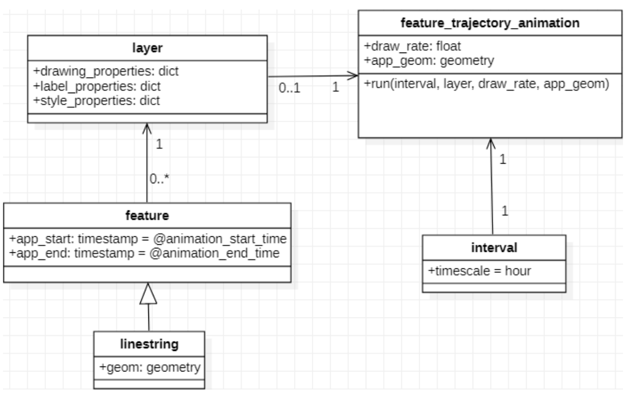


Figure 8. Along the trajectory moving animation UML class diagram.

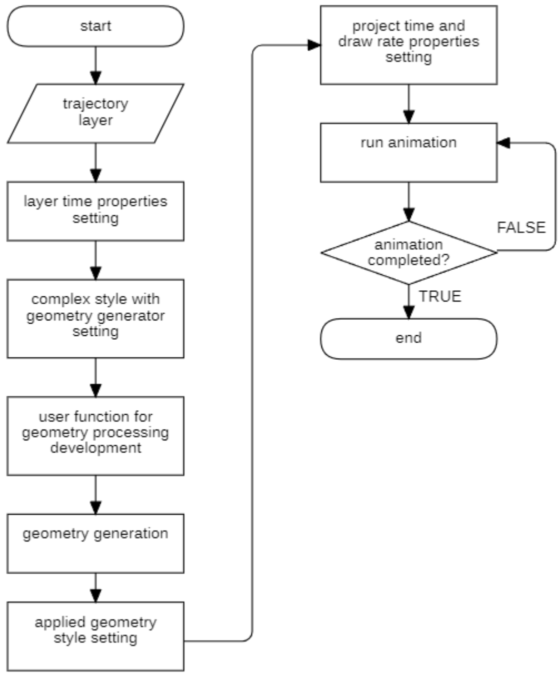


Figure 9. UML block diagram of the algorithm for creating an along the trajectory moving animation.

Animation of object geometry change described as a set of fixed states recorded in a database (in an attribute table) is applicable both to linear and polygonal objects, but assumes using of different algorithms. Animation of linear geometry change is performed using an equal to the along the trajectory moving algorithm – through the gradual drawing of the required number of nodes. The target object is the trajectory itself, when applying moving along a trajectory animation, while the geometric difference between the object states calculated using the QGIS built-in spatial difference function appears to be a target object when animating shape change.

To create an animation of polygonal object shape change, which implements a polygon spreading effect over the time, a unique algorithm (Fig. 10) is required using a set of state describing polygons. The animation is performed using the following list of operations:

1. Setting the Redraw Layer Only option in the Temporal panel of the Layer Properties window for a layer of polygonal objects used to describe object states, if the database contains only the drawing sequence number attribute, or the mode for rendering by date and time, if the time marks are recorded in the database
2. Creating a complex object style using Geometry Generator tuned in the Layer Properties window of a polygonal layer
3. Creating custom geometry processing functions for intermediate shapes generation
4. Embedding of the geometry processing functions as a Geometry Generator equations
5. Setting the animation starting and ending time, step, and the number of frames per second

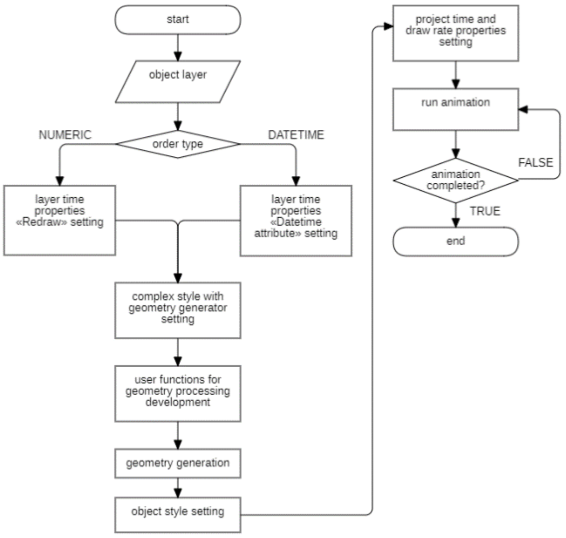


Figure 10. UML block diagram of the algorithm for polygonal object geometry animation based upon an object state database.

Polygonal object geometry animating assumes creating two custom functions in Geometry Generator equation. The first one is receiving input data from the QGIS project and returning the necessary geometry according to a specified time, while the second provides direct calculations on this geometry according to a specified time.

Animations of position and shape change based on the modeling functions were created using similar geometry generation algorithms used to animate position and shape change according to object state database (Fig. 11). A trajectory of the object movements have to be emulated, and a function that returns current position on the emulated trajectory at a given time have to be developed to ensure position change animation. To ensure shape change animation, a list of object geometries is calculated for the full animation interval at once.

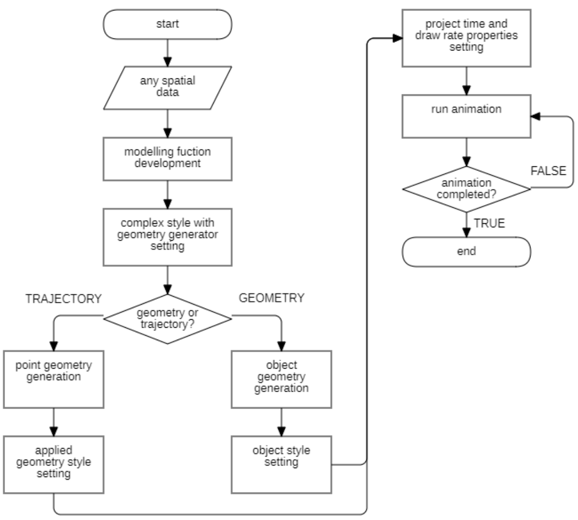


Figure 11. UML block diagram of the algorithm for animation based upon a modelling function.

The key difference is that in this case, new geometries can be synthesized basing on any kind of source data, while in the previous cases the source data were composed of objects of the same geometry type to the target ones. For example, to create a flood animation, a raster of a digital terrain model can be applied as the source dataset, and used to spread polygonal geometry. Similarly, when animating the movement of an object in space, the source data may be composed of the beginning and ending coordinates, and the surface shape on which the object moves.

An animation of a position change based on a modeling function, for the aircraft movement example, that moves from the starting point to the ending (a sphere is used as the movement surface), was created using the following list of operations:

1. Creating an emulating function based on the source data
2. Creating a complex object style using Geometry Generator tuned in the Layer Properties window of a point layer
3. Geometry generation based on created functions
4. Setting the animation starting and ending time, step, and the number of frames per second

In the case of movement along the sphere, the shortest distance between two points is orthodromy, so the orthodromy equation was used in this case for the point marker position calculation at the needed time.

In whole, we developed a set of animation algorithms implementing all the animation types classified in the section 2 of the paper. The whole class diagram showing animation algorithms’ dependencies in QGIS environment is presented in Fig. 12.

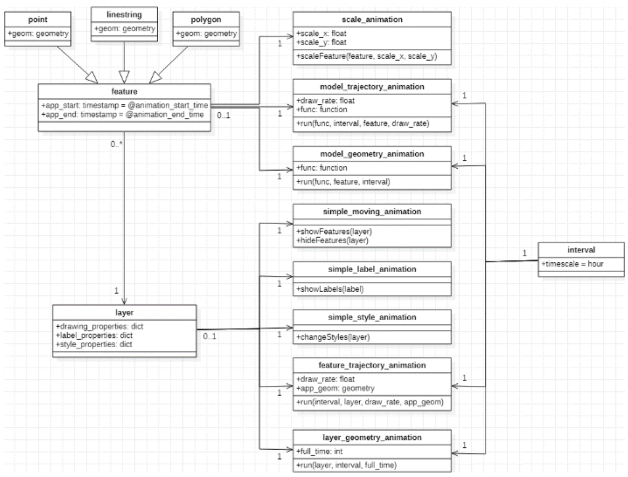


Figure 12. Whole QGIS animation UML class diagram.

Developed algorithms have implement the following methods (taking into account that everyone uses time management parameters):

1. Time-only-based animation – implemented by an algorithm for simple position change animation based on attributes of map objects
2. Affine-transformation-based animation – implemented by an object scaling algorithm
3. Animation based upon object style change – implemented by algorithms for object label, color and transparency change
4. Animation based upon an object geometry generation – implemented by algorithms mentioned in the 5-8 items of animation typology.

## 4. Conclusions

In a result of the research, we developed a number of methods and corresponding algorithms to ensure the implementation of a full cycle animated mapping of spatial object dynamics in a universal desktop GIS environment.

The developed algorithms were embedded in QGIS and documented using class and block diagrams in UML, which makes it easier to port them to other desktop GIS.

During the research, we analyzed existing methods used for creating cartographic animations of objects, and QGIS functionality for applied to create cartographic animations of objects. Finally, we identified the typology of cartographic animations.

Additionally, we analyzed the complexity and ambiguity of the entities and terminology of the discovered domain, differences in similar concepts, and possible discrepancies. As a result of the analysis, generalization and definition of the developed methods for creating animations in GIS, the place of development in the knowledge system of cartography and geoinformatics as the part of GIS-based mapping was determined.

Gained results demonstrate a series of animated mapping examples applicable to produce different types of educational (and demonstrational) visualizations. Proposed typology ensures selection of tool and technological chain for animated map compilation due to educational need and according to the nature of phenomenon discovered in educational process.

## References

Beresnev, A., Semenov, A., Panidi, E., 2022. Hexagonal grids applied to clustering locations in W. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2022, 435-440, doi:10.5194/isprs-archives-XLIII-B4-2022-435-2022

Berlyant, A.M., 1997. *Geoinformational mapping*. Astreya, Moscow. 64 p. (in Russian)

Berlyant, A.M., 2006. The theory of geoimagery. GEOS, Moscow. 262 p. (in Russian)

Berlyant, A.M., Ushakova, L.A., 2000. *Cartographic animations*. Nauchny mir, Moscow. 108. (in Russian)

Bertin, J., 1967. *Sémiologie graphique: les diagrammes, les réseaux, les cartes*. Mouton. 431 p. (in French)

Campbell, C.S., Egbert, S.L., 1990 Animated cartography: Thirty years of scratching the surface. *Cartographica*, 27(2), 24-46. doi:10.3138/V321-5367-W742-1587

DiBiase, D., MacEachren, A.M., Krygier, J.B., Reeves, C., 1992. Animation and the role of map design in scientific visualization. *Cartography and geographic information science*, 19(4), 201-214. doi:10.1559/152304092783721295

Harrower, M., 2003. Tips for designing effective animated maps. *Cartographic Perspectives*, 44, 63-65. doi:10.14714/CP44.516

Harrower, M., Fabricant, S., 2008. The role of map animation for geographic visualization. In: Dodge, M. *Geographic Visualization: Concepts, Tools and Applications*. Chichester, UK: Wiley, 49-65. doi:10.1002/9780470987643.ch4

Lisickij, D.V., Horoshilov, V.S., Kolesnikov, A.A., 2014. Animated cartography – the nature, characteristics and perspectives of Cartography. *Izv. vuzov. Geodeziya i aehrofotos"emka*, 4(C), 91-97. (in Russian)

MacEachren, A.M., 2004. *How maps work: representation, visualization, and design*. Guilford Press, New York. 513 p.

Morrison, J.L., 1974. A theoretical framework for cartographic generalization with the emphasis on the process of symbolization. *International Yearbook of Cartography*, 14(1974), 115-27.

Sidorina, I.E., Pozdnyakova, N.A., Panidi, E.A., Andreeva, T.A., Litvinova, M.V., 2019. Integration of traditional and modern methods in GIS-based mapping. *InterCarto. InterGIS*, 25(1), 35-46. doi:10.35595/2414-9179-2019-1-25-35-46 (in Russian)

Whaley, K.A., 1997. *Production of an animated cartography sample for teaching and instruction, with musings on the nature and usefulness of animated maps*. Graduate Project. Oregon State University, Corvallis. 40 p.