Advances in Aircraft and Spacecraft Science, *Vol. 10, No. 6 (2023) 545-554*
https://doi.org/10.12989/aas.2023.10.6.545

Spatial database architecture for organizing a unified information space for manned and unmanned aviation

Maksim Kalyagin* and Yuri Bukharev[a]

Moscow Aviation Institute, Volokolamskoe Highway 4, 125993 Moscow, Russia

(Received December 8, 2023, Revised January 26, 2024, Accepted February 11, 2024)

Abstract. The widespread introduction of unmanned aircrafts has led to the understanding of the need to organize a common information space for manned and unmanned aircrafts, which is reflected in the Russian Unmanned aircraft system Traffic Management (RUTM) project. The present article deals with the issues of spatial information database (DB) organization, which is the core of RUTM and provides storage of various data types (spatial, aeronautical, topographical, meteorological, vector, etc.) required for flight safety management. Based on the analysis of functional capabilities and types of work which it needs to ensure, the architecture of spatial information DB, including the base of source information, base of display settings, base of vector objects, base of tile packages and also a number of special software packages was proposed. The issues of organization of these DB, types and formats of data and ways of their display are considered in detail. Based on the analysis it was concluded that the optimal construction of the spatial DB for RUTM system requires a combination of different model variants and ways of organizing data structures.

Keywords: aeronautical information; integration of unmanned aircraft systems; Russian Unmanned aircraft system Traffic Management (RUTM); spatial information database; unmanned aerial vehicle (UAV); vector object

1. Introduction

With the appearance of unmanned aerial vehicles (UAV), the need arose to organize a unified information space, allowing external UAV pilots and civil aviation pilots to have full information about each other's actions in real time. Although the main conceptual problems have already been solved (airspace classification, UAV classification, risk assessment methodology, risk-oriented approach to flight clearance practice, etc.), but the issue of organizing a common information space based on modern information technologies remains open (Gupta *et al.* 2016, Zeng *et al.* 2019, Mozaffari *et al.* 2019, Vinogradov *et al.* 2020). In particular, Unmanned aircraft system Traffic Management (UTM) system being created under the aegis of the USA Federal Aviation Administration (Nasa.gov 2020, Icao.int 2023), is intended only for coordination of UAV flights in areas where the presence of manned aircrafts is not expected. Similar systems are being created in China (Unmanned aircraft system aviation Operation Management System (UOMS)) and in

*Corresponding author, Professor, E-mail: mukalyagin@mail.ru
[a]Professor, E-mail: yuribukharev@mail.ru

http://www.techno-press.org/?journal=aas&subpage=7 ISSN: 2287-528X (Print), 2287-5271 (Online)

Europe (U-Space) (Unmannedairspace.info 2018, Lappas *et al.* 2020).

Russia also considers the integration of UAV into the national airspace (NA) on the basis of the concept of joint aeronautical and information services for unmanned and manned aviation. Within its framework, the users of NA should be provided with the necessary information services, which will allow automatically allocating sections of NA for UAV, to observe safe intervals between flights and to prevent conflicts in the air, which in turn will create conditions for safe and efficient flights of unmanned and manned aviation in the unified airspace. One of the basic principles of the concept is the preferential use of electronic component base, software and hardware complexes and software (digital platforms) of Russian design and production when solving the issues of integration of unmanned aircraft systems into the common airspace of the Russian Federation. The purpose of this article is to develop DB architecture of spatial information, i.e., any data with direct or indirect reference to a certain location or geographical area, of RUTM project (Unmannedairspace.info 2020, Uavprof.com 2021), which meets the requirements of the concept.

2. Theoretical basis

The process of DB design is an iterative process and can be refined as the software product develops (Weinberg *et al.* 2010). One of the most important stages of such design is the stage of logical design, which selects the way of data organization, i.e., the process of creating a general information model, independent of the type of real DB management system and other physical conditions, based on data about individual users.

The list of tasks solved by DB of spatial data is determined by the operational capabilities of RUTM users at various phases of UAV flight: out, flight, and after flight. At the same time, DB must support the interfaces required by users at all of the above phases. In accordance with this, the following DB functionality can be distinguished, which it should provide:

- generation and display of two-dimensional and three-dimensional images (Toriwaki and Yoshida 2009) based on spatial data,
- managing the parameters of the territory model display on the graphical client using the application programming interface (API) provided,
- managing the display by means of user interaction devices with the system,
- providing access to a single DB using the network protocol,
- receiving, processing and displaying incoming data from external systems (object traffic, weather conditions, video streams and results of calculation modules).

At the same time the software shall provide the following types of operations with the unified DB information:

- editing spatial data using third-party editor in the source DB (optional),
- editing other data using a third-party editor in the source DB (optional),
- editing of the display settings data in the corresponding DB (a part of the single DB) with the help of the display settings editor,
- converting vector spatial data into an internal format by means of a vector data editor and placing it to a vector DB (a part of the unified DB) by means of a vector data server,
- editing of vector spatial data from vector DB by means of vector data management server and vector data editor (optional),
- forming of a tile package (a part of the unified DB) by means of the tile package generator,
- providing of the necessary tiles on demand of the local clients via visual tiles service,

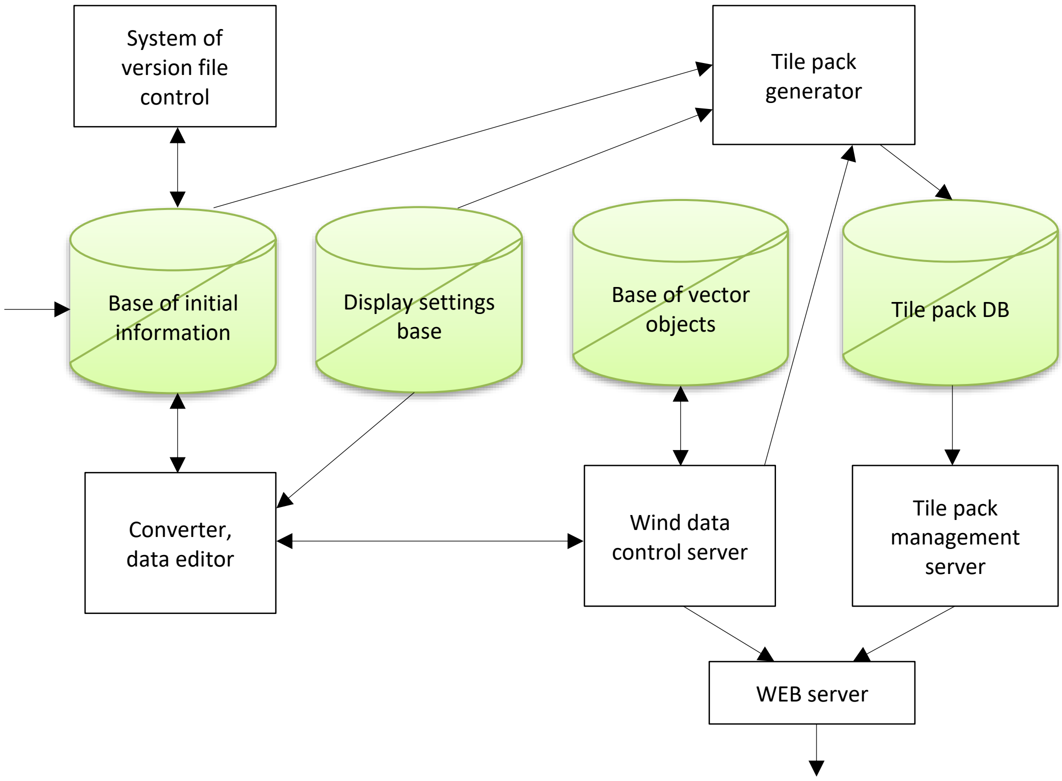

Fig. 1 Structure of the spatial information DB

• providing vector spatial objects to local clients on demand by means of the vector data management server,
• receiving and processing network requests from external clients by means of WEB-server,
• acquisition of spatial and other data by graphic clients and final image generation,
• exporting information from the unified DB to external formats using a data converter.

Taking into account the above information, as well as restrictions on the choice of hardware and software in the technical specifications for the development work on the development of RUTM DB, the authors chose the following DB architecture, presented in the Fig. 1.

Spatial information DB contains the following specialized data modules:

• base of source information necessary for organization of accumulation and storage of spatial and other data from external sources/producers in original formats,
• DB of display settings, which is required for storing and updating 2D/3D spatial information conversion and display settings,
• base of vector objects necessary for effective organization of work with large quantities of vector data (creation, updating, processing and application),
• tile package DB, which enables efficient preparation of raster and vector spatial data for real-time application in client-server architecture, which reduces the risk of data loss and decreases the requirements to the client device functionality.

In addition to the above-mentioned DB the system contains the following special software packages:

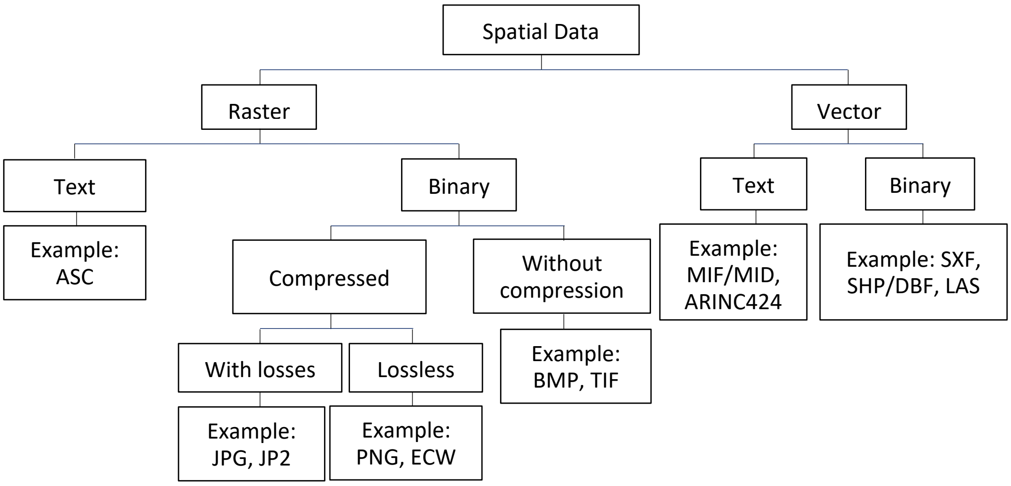

Fig. 2 Example of source information file types

• vector object DB management server, which serves to update, search and transfer vector objects,
• tile package management server, which provides tile search and transfer,
• converter, data editor, allowing to import/export data, create, edit and process vector objects,
• tile packet generator, which serves for creation of tile packets,
• WEB-server, which is necessary for passing the tiles and vector objects.
Below there is the description of the basic modules of the spatial information DB.

3. Methodology

3.1 Structure of the ınitial ınformation base

Source information is spatial, aeronautical, topographic and meteorological information, which is supplied by appropriate developers and providers according to certain standards and in certain formats. In particular, the source information may be arranged by the provider as one file (e.g., aeronautical information) or several files (e.g., layers of vector objects) or thousands of files (e.g., digital elevation model of the territory). The purpose of the source information DB is structured (ordered) storage of the whole range of source data from external sources in original formats without any conversion and in various coordinate and projection systems.

The format of source information files is determined by the provider and can be, as it is shown in the Fig. 2, text vector (e.g., ARINC424 or OSM) or binary vector (e.g., SXF or LAS), text raster (e.g., ASC) or binary raster (e.g., TIFF).

We consider the main types and corresponding formats of source information:

• aeronautical information, the structure of which is defined by ARINC standard, which specifies the electrical interface and protocol for digital data transmission (SAE ITC 2023), and it is provided in such formats as ARINC 424-15, ARINC 424-20, ARINC 424-21, ARINC 816, etc.,
• structure of meteorological information is defined by Weather Information Exchange Model

Table 1 Comparative characteristics of centralized and distributed version control systems

Feature	Centralized version control system	Distributed version control system
File storage	Single file server	Each user has a local file server
Storage access	Requires access for each user	Users can work without a network, but need Internet to share files
System usability	Limited number of users have access to the server	Number of users is not limited
System examples	Users are hosted on a single site	Suitable for small and large projects

(WXXM) standard (Moosakhanian and Hart 2013), designed for the dissemination of weather data, it is presented in the formats WXXM 2.0 and IWXXM 3.0,

• Earth remote sensing materials in the form of orthophotomaps (digital transformed images of the terrain or object) (Bolstad 2016) can be presented in the formats GeoTIFF, ECW, IMG, GeoPDF, etc.

Besides the specified types of information, DB should contain files of regular global digital elevation model (GeoTIFF, HGT, IMG, etc.), files of irregular global digital elevation model or laser survey point clouds (LAS, DXF, CSV, etc.), files of vector objects (SXF, SHP, OSM, etc.), and also files of appropriate objects libraries (3D models, 2D images and coefficient DB). We can distinguish STP (universal format for representing data of 3D models), STL (universal format for 3D models, originally created by albert consulting group for 3D Systems), 3DS (binary file format, used for programs 3D Studio Max 3D), FBX (file format developed by Kaydara, currently owned and still being developed by autodesk), etc. of 3D model file formats.

To register changes in source data, it is necessary to use a version control system, which may be local, centralized or distributed (Zolkifli *et al.* 2018). Local systems, the example of which is Revision Control System (RCS), are simple, but allow working with only one file and have an inconvenient mechanism for multiple users.

There are currently several variants of a centralized version control system: Concurrent Versions System (CVS), Subversion (SVN) and Perforce Revision Control System. SVN is an improved version of CVS, which takes into account atomicity of changes, recording changes, renaming and moving files and directories are subject to version control. Centralized version control systems have many advantages, but at the same time, the centralized server has been shown to be the weak point of the system, as developers lose all data if the main server fails.

Distributed systems of version control, such as Mercurial, Bazaar, BitKeeper, allow leveling out this disadvantage of centralized systems and store data in the local storage of each user. In this case, changes of files are registered only on the local computer and do not affect other projects. A comparative evaluation of centralized and distributed version control systems is presented in the Table 1.

Thus, it seems appropriate to organize DB of the original information in file form with a tree structure of directories. This organization will allow direct accessing to the information, which will solve the problem without the use of DBMS. Metadata is presented in a relational table with links to the data.

3.2 Vector objects DB structure

Vector spatial information, which is a way to represent real world objects in geographic

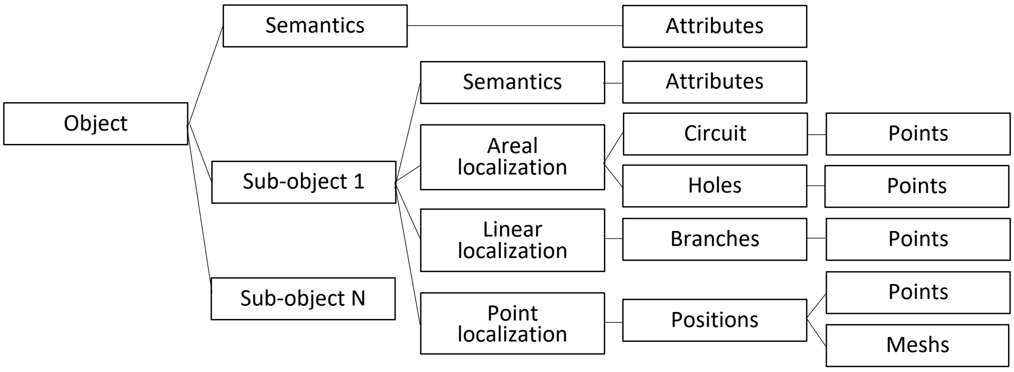

Fig. 3 Example of vector object structure

information system, is (Salton *et al.* 1975):

• metric as a set of contours or lines from coordinate points,

• semantics as a set of attributive data describing an object.

Vector description allows creating compact objects (e.g., buildings or helipads) and very long objects (e.g., sea shoreline or civil aviation route networks). However, when the number of vector objects increases up to hundreds of thousands and millions of units (for example, when describing the whole surface of the planet or its part in a very detailed way), the problem of organization of storage, search and selection of objects arises. Just a direct list of objects becomes an ineffective form of storage.

An effective way to describe such data structures are Bound Volume Hierarchy (BVH) trees (Gu *et al.* 2013), which describe a set of geometric objects as a hierarchy of their framing rectangles (in the case of 2D). Such a hierarchical structure allows efficiently searching and selecting objects by coordinates or visibility area to solve mapping or analytical processing problems.

The proprietary information DB module development seems to be an effective and feasible solution to provide full control over the key part of the spatial DB. Universal vector object structure providing completeness and coherence of information from any external format is presented in the Fig. 3.

For scalability of the system, it is reasonable to organize the vector spatial information base by territorial feature. All objects corresponding to the same location should be grouped in one file. For example, the entire list of vector information relating to a certain settlement will be grouped in one file. And when we change the display scale, the way these data are visualized changes.

3.3 Structure of the object display settings DB

The presence of DB of vector objects implies the need to display them in 2D and 3D modes. The problem is not unambiguous and has multiple solutions, which can be represented as settings of objects displaying parameters and executed as a separate DB module.

For example, parameters for 2D-view could be filled color or texture, thickness, color or texture of outline line, symbolic images of point objects (raster and vector), type, color and size of font, caption template, caption position, objects visibility from scale (generalization) and relative height of objects layers.

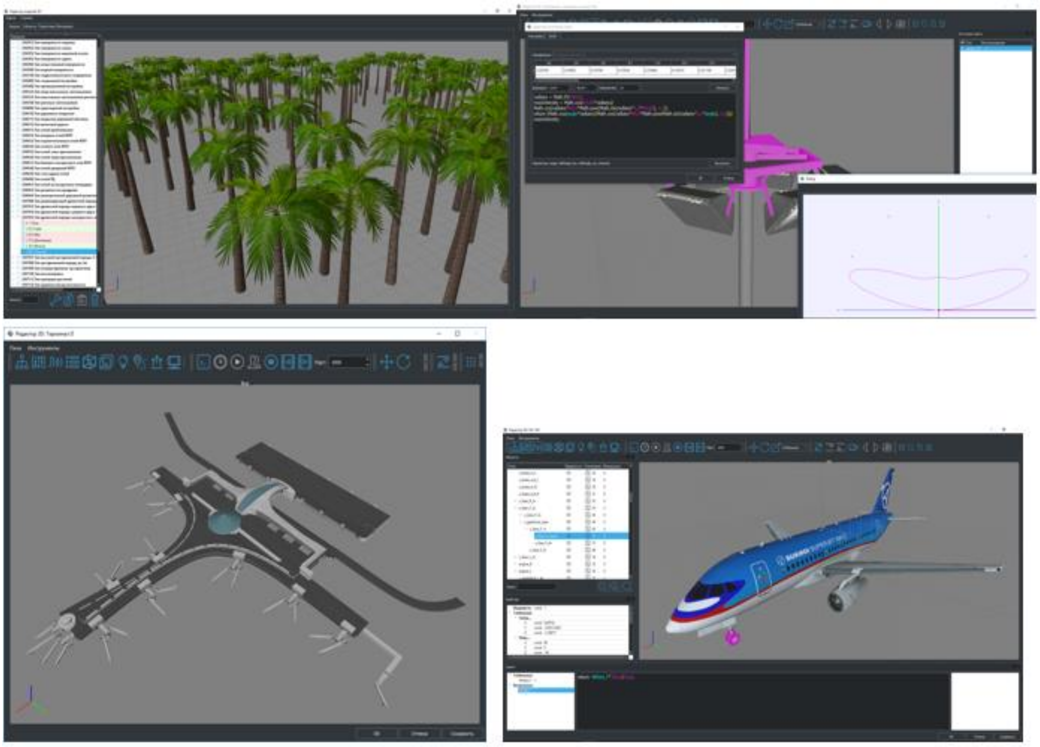

Fig. 4 Example of modeling techniques for different objects

Parameters for 3D-image are modeling techniques of territories (roads, markings, soils, fields, forests, etc.), modeling techniques of water spaces, modeling techniques of procedural buildings and structures, 3D models of unique buildings and structures, modeling techniques of light sources, 3D models of moving objects. These modeling techniques contain numerical modeling characteristics (e.g., curb width, average tree height or water shade) and a set of materials. Each material can consist of several channels (primary color, normal, reflection degree, micro-relief, temperature, etc.) containing appropriate textures and parameters. An example of modeling techniques is shown in the Fig. 4: a) model of scaffolding, b) model of light sources, c) model of unique buildings and structures, d) model of moving objects.

Considering the above, DB of object display settings can be implemented in a single file with a structured format.

3.4 Structure of the tile packet DB

For effective and secure use of spatial data in a client-server architecture through open networks, it is possible to use an approach with the formation of several (color, height, object composition, distribution masks, service masks, etc.) synchronized between each other hierarchical structures of raster and vector tiles.

As a hierarchical structure, it is advisable to use the Quadtree principle (Eberhardt *et al.* 2010, Ong *et al.*2001), according to which at the uppermost (least detailed) level, the terrain (country, region, city, etc.) is represented in a projection that ensures rectangularity of the image (cube, octahedron, etc.). Then the image of the upper level is divided into four quarters, increasing the resolution of the image by a factor of two while maintaining the pixel size. And so on until a given limit of detail per pixel is reached. It takes about 25 levels to achieve centimeter resolution. At that, all raster tiles, i.e., square raster images of equal size, will have the same pixel size, which allows organizing their storage on disk quite easily. Each tile of any data type has a unique identifier, uniquely positioning it in space. Vector tiles contain approximately the same balanced amount of data, as more detail reduces the number of objects to be marked, and large objects are trimmed in their metric on the contour of the tile. Each tile has a unique identifier, which contains information about the immersion level and coordinates (XY) within its level.

The content of raster tiles (color or monochrome masks) is an image in one of the standard formats (JPG, PNG, J2K) or in OpenGL texture format (possibly with compression).

Content of height tiles (height or other physical value) is an image in TIFF format or in a specialized format with compression (e.g., with subtractions).

Vector tiles contents are the sets of objects or parts of objects that have fallen on the area covered by the tile, with thinned metric points according to the scale and relative coordinates, recalculated to the tile coordinate system. A tile also carries information (resources) for displaying all its objects in 2D (MapBox specification for 2D mode) or 3D (3D Tiles specification for 3D mode) modes.

DB of tile packages can be organized by a file structure, organized according to data types. In this case, each file corresponds to a specific tile.

4. Results

Based on the functionality and types of work to be provided by the spatial information DB of RUTM project, its architecture was selected, which includes the following main elements:

- base of source information required for organization of accumulation and storage of spatial and other data from external sources/producers in original formats,
- DB of display settings, which is required for organization of storing and updating 2D/3D spatial information conversion and display settings,
- base of vector objects necessary for effective organization of work with large quantities of vector data (creation, updating, processing and application),
- base of tile packages, enabling efficient preparation of raster and vector spatial data for real-time application in client-server architecture,
- special software packages (vector objects base management server, tile packets management server, converter, tile packets generator and WEB-server).

The tasks of each of these DB, the types of information stored in them (spatial, aeronautical, topographic, meteorological, vector, etc.) and the formats (ARINC, WXXM, GeoTIFF, ECW, IMG, etc.) in which it should be represented are formulated.

It is shown that the optimal construction of the spatial DB for RUTM system requires a combination of different model variants and ways of organizing data structures. In particular, it is advisable to organize DB of source information in file form with a tree-like structure of directories, the base of vector spatial information-on a territorial basis, and all objects corresponding to the same place should be grouped in one file, the base of settings of objects display-in one file with a structured format, the base of tile packages-based on a file structure arranged in accordance with the types of data.

5. Discussion

The existing DBMS have an excessive number of functions that are not necessary for the organization of spatial data, but require refinement of functionality to ensure the unique functionality of RUTM, for example, specific 3D mapping.

Given this circumstance, as well as political risks, namely the introduction of possible sanctions and bans on the use of American and European DBMS, it seems advisable to develop own specialized comprehensive solution for the organization of spatial DB, consistent with the requirements of RUTM.

6. Conclusions

The presented architecture of the spatial information DB of RUTM project makes it possible to provide a unified information field for manned and unmanned aviation by storing various types of information required for flight safety organization and providing it promptly to users in an optimal form. Such architecture requires combination of different variants of models and ways of data structures organization and allows predominantly using software and hardware complexes of Russian production, which minimizes sanctions risks.

References

Bolstad, P. (2016), *GIS Fundamentals: A First Text on Geographic Information Systems*, Eider Press, White Bear Lake, Minnesota, USA.

Eberhardt, H., Klumpp, V. and Hanebeck, U.D. (2010), "Density trees for efficient nonlinear state estimation", *2010 13th International Conference on Information Fusion*, July.

Gu, Y., He, Y., Fatahalian, K. and Blelloch, G. (2013), "Efficient BVH construction via approximate agglomerative clustering", *Proceedings of the 5th High-Performance Graphics Conference*, July.

Gupta, L., Jain, R. and Vaszkun, G. (2016), "Survey of important issues in UAV communication networks", *IEEE Commun. Survey. Tutorial.*, **18**(2), 1123-1152. https://doi.org/10.1109/COMST.2015.2495297.

Icao.int (2023), UTM-A Common Framework with Core Boundaries for Global Harmonization, ICAO, 3rd Edition. Available online: https://www.icao.int/safety/UA/Documents/UTM%20Framework%20Edition%203.pdf.

Lappas, V., Zoumponos, G., Kostopoulos, V., Shin, H.Y., Tsourdos, A., Tantarini, M., Shmoko, D., Munoz, J., Amoratis, N., Maragkakis, A., Machairas, T. and Trifas, A. (2020), "Eurodrone, a European UTM testbed for U-Space", *International Conference on Unmanned Aircraft Systems*, September. Available online: https://www.researchgate.net/publication/347154818_eurodrone_A_European_UTM_Testbed_for_U-Space.

Moosakhanian, A. and Hart, D. (2013), "Integrating information towards digital ATM-weather information exchange model", *Air Transportation Information Exchange Conference*. Available online: https://slideplayer.com/slide/1447770/.

Mozaffari, M., Saad, W., Bennis, M., Nam, Y. and Debbah, M. (2019), "A tutorial on uavs for wireless networks: Applications, challenges, and open problems", *IEEE Commun. Survey. Tutorial.*, **21**(3), 2334-2360. https://doi.org/10.1109/COMST.2019.2902862.

Nasa.gov (2020), UAS Traffic Management, Conflict Management Model, Available online: https://www.nasa.gov/sites/default/files/atoms/files/2020-johnson-nasa-faa.pdf.

Ong, G.H., Chew, C.M. and Cao, Y. (2001), "A simple partitioning approach to fractal image compression",

Proceedings of the 2001 ACM Symposium on Applied Computing, March.
SAE ITC (2023), ARINC Industry Activities. Available online: https://www.aviation-ia.com/.
Salton, G., Wong, A. and Yang, C.S. (1975), "A vector space model for automatic indexing", *Commun. ACM*, **18**(11), 613-620. https://doi.org/10.1145/361219.361220.
Toriwaki, J. and Yoshida, H. (2009), *Fundamentals of Three-Dimensional Digital Image Processing*, Springer, New York.
Uavprof.com (2021), RUTM1 and UAV integration in general Russian airspace. Astra Satellite Technologies. Available online: https://uavprof.com/wpcontent/uploads/2021/06/presentation.pdf.
Unmannedairspace.info (2018), EU-China APP Drone Workshop Highlights UTM Progress in China. Available online: https://www.unmannedairspace.info/uncategorized/eu-china-app-drone-workshop-highlights-utm-progress-china/.
Unmannedairspace.info (2020), Russia's Nationwide UTM System Passes Technical Design Milestone. Available online: https://www.unmannedairspace.info/uncategorized/russias-nationwide-utm-system-passes-technical-design-milestone/.
Vinogradov, E., Sallouha, H., De Bast, S., Azari, M.M. and Pollin, S. (2019), Tutorial on UAV: A Blue Sky View on Wireless Communication, arXiv preprint arXiv:1901.02306.
Weinberg, P., Groff, J. and Oppel, A. (2010), *SQL. The Complete Reference*, Mcgraw-Hill, New York, USA.
Zeng, Y., Wu, Q. and Zhang, R. (2019). "Accessing from the sky: A tutorial on UAV communications for 5G and beyond", *Proc. IEEE*, **107**(12), 2327-2375. https://doi.org/10.1109/JPROC.2019.2952892.
Zolkifli, N.N., Ngah, A. and Deraman, A. (2018), "Version control system: A review", *Procedia Comput. Sci.*, **135**, 408-415. https://doi.org/10.1016/j.procs.2018.08.191.

AP