Object-Oriented Topological Management System of Spatially-Distributed Databases

Yu. G. Vasin* and Yu. V. Yasakov

Center for Informatics and Intellectual Information Technologies, Institute of Information Technology, Mathematics and Mechanics, State University of Nizhny Novgorod, Nizhny Novgorod, Russia

*e-mail: *yuri961@yandex.ru*

Abstract—This paper discusses the problems associated with the development of the specialized database management system (DBMS) GIS Terra Plus for storage, processing, and analysis of spatially distributed data (SDD). Description models and the presentation structure of the SDD, the scheme of databases of the GIS Terra Plus DBMS, the query system, and operating conditions are considered.

Keywords: database, spatially distributed data, topology, format, data model, object

DOI: 10.1134/S1054661816040180

INTRODUCTION

In comparison with traditional databases, additional problems associated with large amounts of data, data dimensions, increase in the number of spatially logical and topological relations between data elements and data heterogeneity arise in the development and processing of databases that contain geospatial information of a complex structure. This leads to the need to develop models, structures, and data formats that differ significantly from the traditional representation methods [1–6].

While traditional databases can store and process numeric and text information, spatial data have the advanced functionality that makes it possible to store the whole spatial object that combines both traditional data types in the form of narrative or attribute information and geospatial data on the position of the object in space, as well as intraobject and interobject relationships.

Currently, the most common and popular DBMS are based on a relational data model. The relational data is represented by two-dimensional data tables with the changing relationships between different columns. It closely resembles data representation by a human.

To date, a growing number of developers of custom applications for DBMS are faced with the fact that the use of traditional relational models does not meet the requirements for storage and processing capabilities of complex unstructured spatially distributed data and query processing speed for them. Therefore, the major DBMS developers have begun to incorporate support for object orientation, as well as additional spatial data representation and processing tools, into their products. In addition, support tools of uninterpreted (bit) data have become more efficient.

From this viewpoint, the following developments are of interest.

First, consider DBMS Oracle [7] with the Oracle Spatial extension, a DBMS component that provides storage, retrieval, and update of spatial data. Oracle Spatial consists of the circuitry that determines the storage, syntax, and semantics of supported geometrical data types, the spatial indexing mechanism, and the set of operators and functions to execute spatial queries and analytical operators.

DBMS PostgreSQL [8] is a free object-relational database management system, one of the most developed of the open source DBMSs in the world. It is a real alternative to commercial databases. The PostgeSQL expansion PostGIS should also be noted [9]. PostGIS was released in 2001 by Refractions Research. It is free and open-source software and it competes with commercial solutions. The main advantage of PostGIS is the use of SQL with spatial operators and functions. In addition to simple storage of this type of data, PostGIS provides a wide range of operations on them. PostGIS is the extension of the object-relational DBMS PostgreSQL for storing of the geographical data in the database. PostGIS supports spatial indices R-Tree/GiST and geodata processing functions.

DBMS GIS Terra [3–6] is based on a special geospatial data presentation format, an object-oriented topological structured format of the integral file (IF). This DBMS was used in the development and operation of a number of automated mapping systems: AWS RASTR-2 (digitization of topographic maps of the scale 1 : 10000–1 : 1000000), AWS RASTR-2P (digitization of urban plans of the scale 1 : 10000–1 : 25000),

Received May 16, 2016

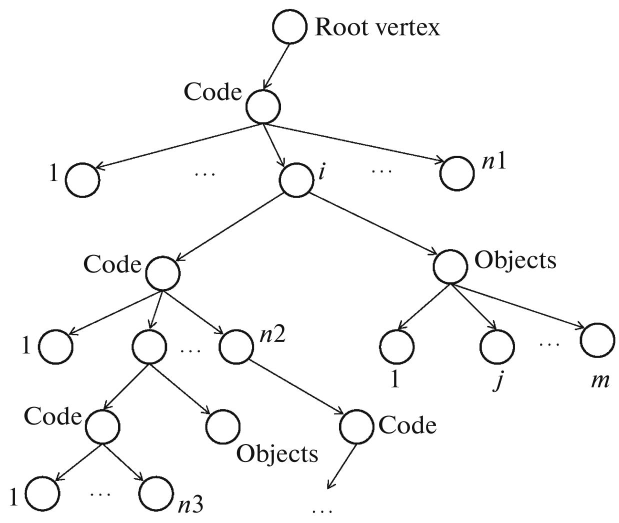


Fig. 1. Tree of database objects.

ASOIMK (digitization of nautical charts and tablets for hydrographic surveys), AKS-LES (digitization of forest management maps correlated with the topography), ARM RASTR-TP (digitization of city maps of the scale 1 : 500–1 : 5000), and in the development of different types of thematic databases (of land plots, floor plans, underground utilities, etc.).

This paper is focused on the problems of the further development of the DBMS Terra GIS based on modern tools for storage and processing of data, taking into account the characteristics of the actual tasks on the representation, storage, and processing of topologically related complex structured geospatial information.

DATA MODEL OF DBMS TERRA GIS

The DBMS Terra GIS is an object-oriented structured topological data model (integral file format). It is maintained in a database, which is a single file of a special structure, that contains information organized in a binary data tree of arbitrary depth. The database integrity is provided by the possibility of conducting a special system log and checkpointing. The database physical structure is a tree of objects constructed in accordance with the hierarchy of the multiposition classification object code of the domain (Fig. 1). In addition, there is the establishment (and automatic support) direct (network) connections between database objects. As a result of this approach, the following features are provided:

• individual quick access to a particular object;

• quick sampling of a collection of objects united by a single generic code;

• sampling of a set of objects united by network links.

The basic information unit of the database is the object. All the information about a particular object is contained in a structured record. This ensures fast reception of any object as a whole (for example, roads with all roadside structures). The object has a dynamic structure adaptable to the needs of specific applications (Fig. 2). The object can be in a database or issued to the user in any combination of its constituent parts.

The database supports several types of objects:

• an elementary object composed of one database record in the integrated file format (binary, line, or areal object);

• a complex or compound object consisting of a number of database records;

• a structured object consisting of a database record containing descriptions of the main object (e.g., roads) and accompanying objects (e.g., bridges and roadside structures);

• a distributed object that combines several types of previous objects from different databases that cover some part of the electronic map.

Particular attention was paid to the implementation of interobject relationships in the geospatial data database. Usually, steady relationships between objects (either via the linkage field or through the interruption field) are always symmetrical. This ensures the efficiency of the search and update of information in the database. However, there are many cases when the symmetrical connections cause only additional maintenance overhead, for example, links to objects of the topographic base (which is constant) from themed objects added to it. In this case, only the link from the theme object to the object/objects of the topographic base is necessary. DBMS tools ensure the implementation of this asymmetrical relationship by their additional semantic description. In this case, the one-way link bit is raised in the logical relationship scale. There are also situations when it is possible to establish a one-way relationship from one object to tens or even hundreds of objects of the same classification code. For example, it is possible to establish such a relationship from the object "Rock" to the set of elementary bitmap objects of which it consists. In this case, in the logical ratio scale the bit's "one-way link" and "multiple links" are raised, and in the relationship attribute field a complex attribute "List of object numbers" is added that contains the attribute list with the numbers of objects by the link. In addition, symmetric multiple links are allowed. There are cases when it is necessary to establish relationships between objects of different databases. They are implemented in the same manner for all types of relationships (connections and interruptions). A complex attribute "Interdatabase link description" is added to the relationship. It contains three simple attributes (one of them is required).

• Database file name (without the directory name).

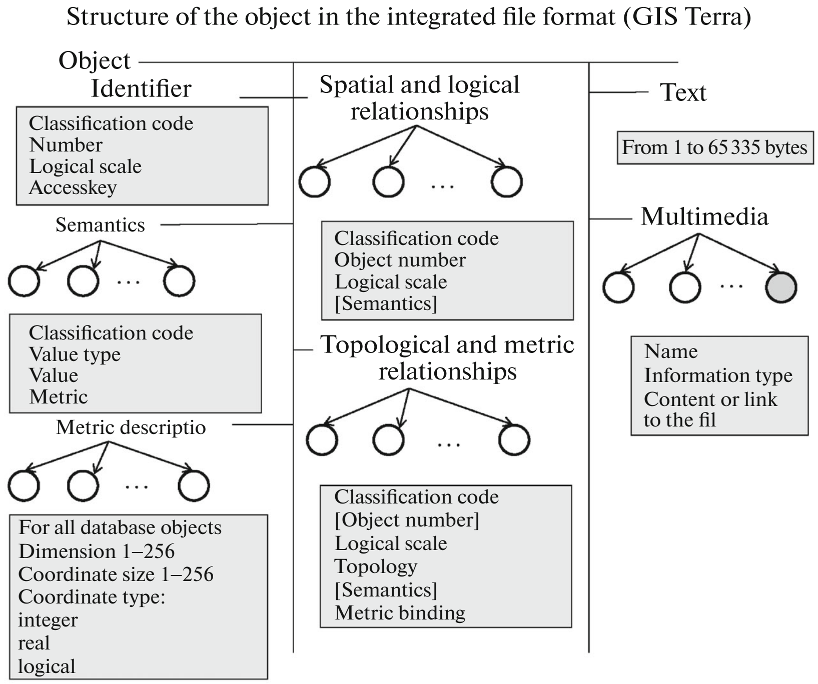


Fig. 2. Structure of the object in the IF format of the GIS Terra.

• Alias name. The desired directory name on the disk can be assigned to this alias. Aliases should be stored in the registry. This attribute can be missing; the base will be searched in the directory where the original database is. The same thing will happen if there are no aliases in the registry.

• Database type (processing format and interchange format). This attribute can be missing; by default, the database in the integral file processing format is assumed.

The bit "Interdatabase link" is added to the logical relationship scale.

Quite often, there are situations when an object in the database should have not only a semantic and metric descriptions, but also the description by a set of other objects. For example, the object "House" on the city map can be a whole database that contains a description of its elements (floors, rooms, communication systems, etc.). In this case, the IF format provides the storage of such a database in the multimedia object field and the automatic switch to it if necessary.

MODEL DEVELOPMENT IN THE DBMS GIS TERRA PLUS

In order to further extend the functionality of the DBMS GIS Terra in the tasks of collection, storage, processing, and analysis of the DWP (support for client-server version, multiversioning of object presentation, possibility of the creation of a common data fields for large areas, etc.), it was decided to implement the object oriented topology data model in the DBMS PostgreSQL [8] with the PostGIS extension [9].

In developing the models, formats, and structures of data for the efficient implementation of object-oriented topological structured model of the IF, it was necessary to solve several fundamentally important problems within the relational DBMS PostgreSQL:

• hierarchical access to database objects;

• multiversion storage of all elements (fields) of an object;

• efficient coding of the metric description of the attributes and objects;

• logging of operations on the object that contains information about update operations on it (on its fields);

• object connection with the source metadata, from which the object was obtained (topographical and marine charts, hydrographic survey maps, etc.);

• execution of SQL queries along with non-SQL queries to the DBMS GIS Terra;

• support of operations on the object (its fields) obtained from the database;

• conversion of information from the DBMS GIS Terra to the DBMS GIS Terra Plus (PostgreSQL), and vice versa.

These problems were solved by developing a specific database schema, which is a series of interconnected PostgreSQL database tables (Fig. 3), and by the use of 'binary' data for complex structured object

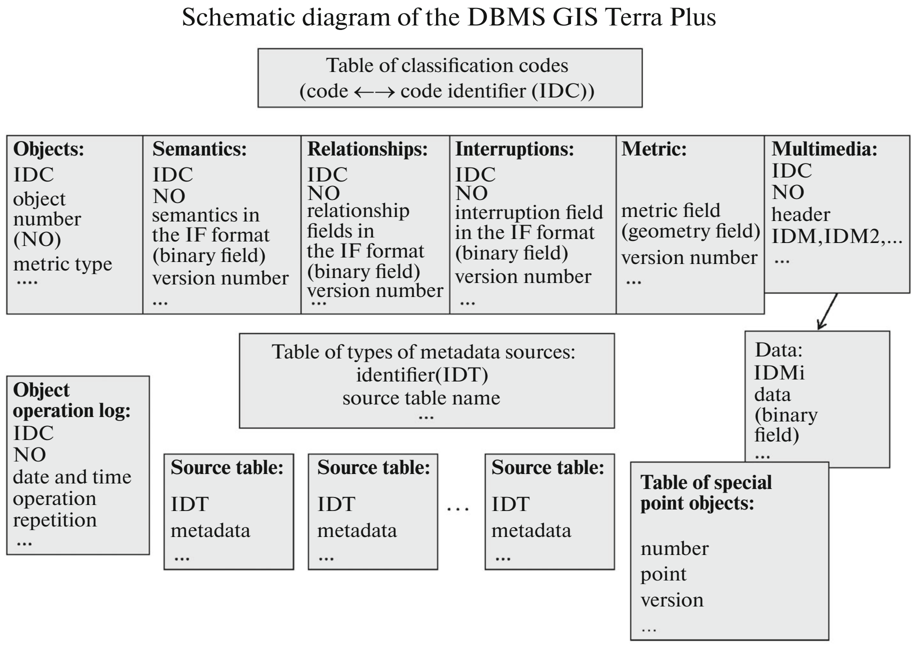


Fig. 3. Schematic diagram of the database.

fields with the development of special software support tools.

In order to implement the hierarchical access to database objects, a special table of classification codes was developed that stores multipositional classification codes of objects (each code position has a separate column) and their corresponding unique identifiers. When executing a query on the selection of objects by the (generalized or full) classification code, unique identifiers of classification codes are sampled from this table, and then the full object is collected from the tables of attributes, relationships, interruptions, metric description, and multimedia information using the obtained unique identifiers.

Each object in the database is identified by a unique identifier of the classification code and the number within this classification code.

Data can be sampled from the database as a separate object and as a group of objects united by the generalized or full classification code.

For each field of the object (semantics, relationships, interruptions, metric, and multimedia), the database has a separate table, in which records corresponding to different versions of the object can be stored. Version no. 0 is the latest version. When executing a query for the selection of objects, it is possible to specify the number of the desired version of the object field for each individual object field.

The field "geometry" is used to store metric descriptions of objects in the metric description table. The DBMS supports two options for with this field. The first involves the access to the object metric through PostGIS functions. This method does not require a physical realization of the "geometry" field, but it requires additional time for the transformation of metric description coordinates from the internal representation to the external, and vice versa. The second option involves direct access to the coordinates of the metric description at the physical layer, which significantly improves performance, but can cause problems in the transformation of the metric description structure (for example, when a new version of PostGIS is released). PostGIS has a rich set of functions for handling the metric description of objects (coordinate transformation, analysis of the mutual arrangement of objects, computational geometry, etc.).

In the DBMS GIS Terra, fields with the semantic description of objects, spatial and logical, and topological-metric relationships between objects have a complex structure and are an arbitrary (not necessarily ordered) set of elements containing the classification codes of objects, arbitrary sets of attributes, coordinate binding, etc. It is impossible to efficiently implement such a design using a relational table. Therefore, these parts of the object data are stored in "binary" fields. DBMS GIS Terra Plus tools can read and write to these fields. Upon receipt of these fields from the database, they are processed by conventional DBMS GIS Terra tools.

Thus, the object model in the IF format of the DBMS GIS Terra was expanded and became as follows (Fig. 4):

IF data models complement each other and make it possible to more efficiently simulate SDD objects.

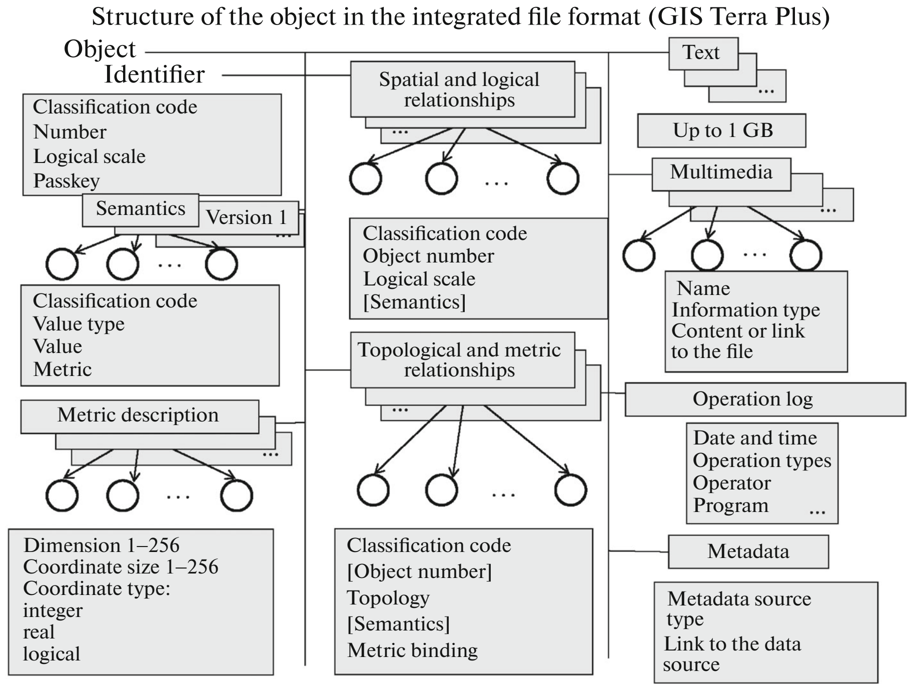


Fig. 4. Structure of the object in the GIS Terra Plus format.

During the DBMS operation depending on the applications used for the SDD processing the database can be converted on the fly from one IF model to another. This provides both new opportunities for the SDD representation and the use of existing applications and software systems for the SDD processing (e.g., vectorization software system PGS and NMK).

QUERY SYSTEM

The developed DBMS GIS Terra Plus supports two modes of operation of the query system:

—structured non-SQL queries to the DBMS GIS Terra [5];

—additional SQL queries;

Non-SQL queries refer to queries for data selection in the fields of attributes, relationships, and object interruptions associated with the specifics of the presentation of these fields in the database. A special library of stored procedures was developed for the execution of such queries. When executing a non-SQL query after the selection of a suitable object, it is further analyzed (namely, fields of attributes, relationships, or interruptions) using a corresponding procedure stored on the server.

Database objects can be selected using additional SQL queries. The main ones are listed below:

• by metric description. In this case, the query for metric description is excluded from is non-SQL query (if it is present);

• by logical object scale. In this case, the query for the logical object scale (if it is present) is excluded from the non-SQL query;

• by multimedia field. In this case, the query for elements of the multimedia field (if it is present) are excluded from the non-SQL query;

• by object source type;

• by source name;

• by source metadata;

• by scale;

• by algebraic conditions on the values of depth marks.

Non-SQL queries and additional SQL queries can be used either separately or simultaneously, which significantly increases the efficiency of the query system of the DBMS. In the case of the simultaneous use, the non-SQL query is analyzed and queries for object's fields, if any in the additional SQL query, are excluded from it.

Here is an example of the non-SQL query for the selection of objects in the query language of the DBMS GIS Terra.

Query: from the database select all objects "Reliable coastline" (classification code 4.1.1), which pass through "Rocky coast" (classification code 2.1.1.1) and "Sand with stones" (classification code 2.5.9.3).

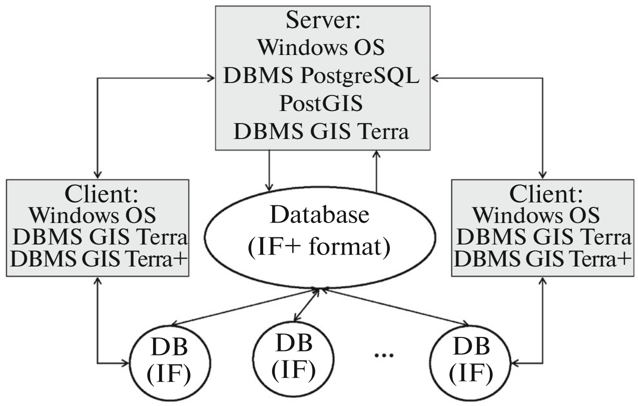


Fig. 5. Architecture of the DBMS GIS Terra Plus.

```
Code=4.1.1
TypeCode=CODE (only the objects of this hierarchy level)
Filter=P((K(2.1.1.1.1.))(K(2.5.9.3.)))
```

The same query to the DBMS GIS Terra Plus will be as follows:

```
SELECT < selected object fields> FROM < list of object tables>
WHERE n=3 AND c1=4 AND c2=1 AND c3=1
AND query_if('P((K(2.1.1.1.1.))(K(2.5.9.3.)))')=true;
```

Here, n is the number of positions of the object classification code (sets the level of the hierarchy of the classification code); c1, c2, c3 are the respective positions of the object code; query_if is the stored procedure of the DBMS GIS Terra Plus that analyzes the object by the provided selection filter.

OPERATING CONDITIONS

The DBMS GIS Terra Plus has a client-server architecture (Fig. 5).

The DBMS server can function under Microsoft Windows or Linux operating system (e.g., Astra Linux Special Edition).

The DBMS PostgreSQL 9.3 or later and PostGIS 2.1 or later (PostgreSQL extension designed for spatial data storage and processing) should be installed on the server.

In addition, stored functions of the DBMS GIS Terra Plus that ensure the execution of queries for selection of database objects by some fields of the object (interruptions, relationships, and characteristics) should be on the server.

The DBMS GIS Terra and GIS Terra Plus should be installed on the client side. Clients operate under Microsoft Windows.

APPLICATION DEVELOPMENT ENVIRONMENT

The developed DBMS is intended for both use within existing applications for the SDD development and for the creation of new applications and software systems. Therefore, one of its main components is the API, a set of predefined classes, procedures, functions, and structures for use in external software products that are represented in the form of corresponding libraries.

It is preferable to develop new applications in the environment of contemporary programming system Embarcadero RAD Studio, a single integrated application development environment for MS Windows, iOS, and Android using components of FireDAC (universal data access library designed for the application development for different devices connected to the corporate databases and providing a highly versatile and native access from Delphi and C++Builder to InterBase, SQLite, MySQL, SQL Server, Oracle, PostgreSQL, DB2, SQL Anywhere, Advantage DB, Firebird, Access, Informix, DataSnap, etc.).

CONCLUSIONS

The DBMS GIS Terra Plus has the following main differences from the existing methods of representation and processing of SDD.

• Supports object-oriented complex structured topological model of SDD.

• All information about a single object is stored in one structured record. This ensures fast reception of any object as a whole (for example, roads with all roadside structures).

• Database object has a dynamic structure that can be adapted to the needs of specific applications.

• Object can store any information (including multimedia).

• No restrictions on the structure of objects in databases, which greatly simplifies the process of representation of real-world objects: there is no need to break them apart to set them in different relationships, as is the case, e.g., in relational databases.

• The total number of objects in the database and the sizes of these objects are limited only by the database size.

• Tools for protection against unauthorized access to the database as a whole and to its objects (down to a single object) provide distributed access to information.

• A key based setting of object attributes is used. Firstly, it makes it possible to have any variable number of attributes in the object and, secondly, to have an arbitrary order of the motion of attributes in the object. This is the difference from data models of most other (recording oriented) DBMSs, in which quite severe restrictions are imposed on the recording structure.

Operation	Number of objects	Object type	Capacity (MB)	Time (s)	Object / sec
Record	3611586	All	650	10800	350
Record	2371735	Depth marks	76	1935	1200
Sample	3611586	All	650	270	13000
Sample	2371735	Depth marks	76	44	54000
Sample	1239851	In addition to depth marks	569	145	8600
Fragment cutting [–81.9 23.8;-78.5 28.7]	137445	All	7	6	21907
Fragment cutting [–81.9 23.8;-80.5 25.7]	62623	All	3	4	15655

Fig. 6. Time characteristics.

• Self-determined form of the presentation of attributes. All attributes contain their own full description including the type of its value.

• It is possible to code recursively complex attributes for a structured description of the multivariable object attributes.

• It can be used as an instrument for the spatial coordinate binding of any attribute that makes it possible to efficiently describe spatially distributed attributes. Each attribute can have its own metric description.

• Supports "Database" characteristics. With it, a situation is implemented when an object in the database should have not only semantic and metric descriptions but also the description by a set of other objects. For example, the object "House" on the city plan can represent a whole database that contains descriptions of its elements (floors, rooms, communication systems, etc.). This database can be stored directly in the object or only a link to it.

• It is possible to establish a spatial and logical and topological and metric relationships between any objects.

• There is an invariance with respect to the subject areas. All interaction with the real world objects in a given domain is carried out using a special domain knowledge base (classifier).

DBMS GIS Terra Plus has the following features:

• hierarchical access to database objects;

• multiversion storage of all elements (fields) of an object;

• efficient coding of the metric description of the attributes and objects;

• logging of operations on the object that contains information about update operations on it (on its fields);

• object connection with the source metadata, from which the object was obtained (topographical and marine charts, hydrographic survey maps, etc.);

• execution of SQL queries along with non-SQL queries to the DBMS GIS Terra;

• support of operations on the object (its fields) obtained from the database;

• conversion of information from the DBMS GIS Terra to the DBMS GIS Terra Plus (PostgreSQL) and vice versa.

The DBMS GIS Terra Plus was tested experimentally for the creation and management of bathymetric database designed to provide users with relevant hydrographic information.

In this domain, application possibilities were shown and the performance of the developed DBMS was estimated. Performance estimation was carried out on the following computer:

• Intel Core i5 processor with a clock speed of 3 GHz;

• RAM 8 GB;

• PostgreSQL 9.3 with the extension PostGIS 2.1 (DBMS server was installed with default settings);

• DBMS GIS Terra;

• Microsoft Windows 7 (64 bit).

The BBD prototype of 1056 electronic navigational charts was loaded. The charts were in the S-57 format, the main format of electronic navigational charts designed for the data exchange between hydrographic services, agencies, and developers of cartographic products and systems.

Time characteristics of the DBMS GIS Terra Plus on operations of recording and sampling of data are shown in Fig. 6.

These time characteristics show that the tools of the GIS Terra Plus provide high-performance of applications for the analysis and processing of the SDD.

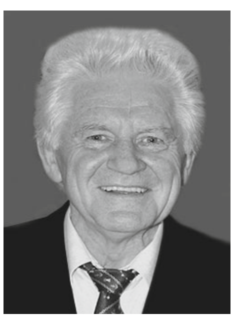
ACKNOWLEDGMENTS

This work was supported by the Russian Science Foundation, project no. 16-11-00068.

REFERENCES

1. Yu. G. Vasin, S. A. Dmitriev, and R. Yu. Kobrin, "Information model for images in the automated systems for processing the graphic information," in *Proc. All-Union Conf. Automated Systems for Image Procession (ASIP-86)* (Lvov, Sept. 1986), pp. 6–8.
2. Yu. G. Vasin, S. A. Dmitriev, and R. Yu. Kobrin, "Information and terminology support of map database," Nauch.-Tekh. Inf. VINITI, NTI Ser. 2 (Moscow, 1989), pp. 20–28.
3. Yu. G. Vasin and Yu. V. Yasakov, "GIS Terra: a graphic database management system," Pattern Recogn. Image Anal. **14** (4), 579–586 (2004).
4. Yu. G. Vasin and Yu. V. Yasakov, "Electronic archive of nautical navigation charts with remote access," in *Proc. 8th Int. Conf. on Pattern Recognition and Image Analysis: New Information Technologies* (Yoshkar-Ola, 2007), Vol. 2, pp. 211–212.
5. *Database Management System Terra. User's Manual* (Lobachevsky State Univ. of Nizhny Novgorod Research Institute for Applied Mathematics and Cybernatics, Nizhny Novgorod, 2004), Book 1 [in Russian].
6. Yu. G. Vasin and Yu. V. Yasakov, "System for controlling the videodata base," in *Interuniversity Collection of Scientific Papers "Methods and Means for Processing Graphical Information"* (Gorky Univ. Named after N.I. Lobachevsky, Gorky, 1989), pp. 93–115 [in Russian].
7. M. Corey, M. Abbey, I. Abramson, and B. Taub, *Oracle8i Data Warehousing* (McGraw-Hill, 2001).
8. R. Stones and N. Matthew, *Beginning Databases with PostgreSQL: from Novice to Professional* (Apress, 2005).
9. PostGIS. User Manual. http://gis-lab.info/docs/postgis/manual/

Translated by O. Pismenov



Yurii Grigor'evich Vasin. Born 1940. Graduated from Gorky State University in 1962. Received a doctoral degree in 1988. Received a title of Professor in 1994. Honored Worker of Science and Technology of the Russian Federation, Honored Worker of Higher School, and Winner of the USSR Council of Ministers. Head of the Center for Informatics and Intellectual Information Technologies of the Institute of Information Technology, Mathematics and Mechanics of the State University of Nizhny Novgorod. Scientific interests: theoretical and applied computer science, pattern recognition and image processing, and information technology. Author of more than 150 papers.

Yurii Vasil'evich Yasakov. Born 1949. Graduated from the Faculty of Computational Mathematics and Cybernetics of the Gorky State University in 1971. In 1971, joined the Institute of Applied Mathematics and Cybernetics of the of the Gorky State University. Senior Researcher at the Center for Informatics and Intellectual Information Technologies of the Institute of Information Technology, Mathematics and Mechanics of the State University of Nizhny Novgorod. Winner of the USSR Council of Ministers in the field of cybernetics. Author of more than 30 papers. Scientific interests: system programming, database, information technology, digital cartography.