УДК 004.65
DOI: 10.26102/2310-6018/2023.41.2.002

Situation-oriented databases: backing up virtual multi-documents of dynamic data processing objects model

A.S. Gusarenko

Ufa University of Science and Technology, Ufa, the Russian Federation
gusarenko.as@ugatu.su✉

Abstract. In database management systems, since the creation of user applications, the problem of data backup has not lost its relevance. With the development of technologies in the field of Internet programming, cloud data backup methods have appeared. Cloud-based backups are gaining ground in the information technology space. Situation-oriented databases (SODBs) at the current stage need their own backup tools. As part of the microservice architecture, since heterogeneous sources and results of data processing in the SODB are taken out of the local infrastructure, it is required to use modern backup capabilities. First of all, it is necessary to reserve virtual data arrays collected from virtual multi-documents as well as dynamic data processing objects. In SODB, multi-documents and dynamic data processing objects are the main elements involved in data manipulation; their content is heterogeneous data sources, intermediate processing results and the final processing result before uploading to the data receiver. It is proposed to solve this problem using a situation-oriented approach by adding a backup model, as well as developed algorithms for backup and operating cloud disks and cloud storages. Previously, the issues of backup in SODB were not given due attention because the model assumed the use of the current state memory mechanism, which guaranteed the protection of data from possible damage and a return to the previous processing steps was provided by editing it. In addition, each state of the model provided for error handling that occur during processing. With the growing need for redundancy of external heterogeneous sources, new equipment is required to eliminate gaps in the backup implementation of SODB. This kind of equipment has not been suggested before; this paper discusses its implementation, and a prototype of the SODB software, accompanying the process of course design in "Databases" course, is used.

Keywords: situation-oriented database, built-in dynamic model, heterogeneous data sources, backup, virtual multi-documents, dynamic data processing objects, RESTful-services.

Acknowledgments: this research is supported by RFBR (grant 19-07-00682); the results of the study, reflecting the structure of the developed software solution, have been obtained as part of the state assignment No. FEUE-2020-0007.

For citation: Gusarenko A.S. Situation-oriented databases: backing up virtual multi-documents of dynamic data processing objects model. *Modeling, Optimization and Information Technology*. 2023;11(2). URL: https://moitvivt.ru/ru/journal/pdf?id=1322 DOI: 10.26102/2310-6018/2023.41.2.002

Ситуационно-ориентированные базы данных: резервное копирование виртуальных мультидокументов модели динамических объектов обработки данных

А.С. Гусаренко

Уфимский университет науки и технологий,Уфа, Российская Федерация
gusarenko.as@ugatu.su✉

Резюме. В системах управления базами данных с момента создания пользовательских приложений не потеряла актуальность задача резервного копирования данных. С развитием технологий в русле интернет программирования появились облачные методы резервирования

© Гусаренко А.С., 2023

данных. Резервное копирование, основывающееся на облачных хранилищах, занимает все более уверенные позиции в сфере информационных технологий. Ситуационно-ориентированные базы данных (СОБД) на текущем этапе нуждаются в собственных средствах резервного копирования. В рамках микросервисной архитектуры, поскольку гетерогенные источники и результаты обработки данных в СОБД выносятся за пределы локальной инфраструктуры, требуется задействовать современные возможности резервного копирования. Прежде всего требуется резервировать виртуальные массивы данных, собранные из виртуальных мультидокументов, а также динамические объекты обработки данных. В СОБД мультидокументы и динамические объекты обработки данных являются основными элементами, участвующими в оперировании данными, их содержимое – это гетерогенные источники данных, промежуточные результаты обработки и конечный результат обработки до выгрузки в приемник данных. Данную задачу предлагается решить с использованием ситуационно-ориентированного подхода, добавляя модель резервного копирования, а также разработанные алгоритмы резервного копирования и работы с облачными дисками, облачными хранилищами. Ранее вопросам резервного копирования в СОБД не уделялось должного внимания, так как модель предполагала использование механизма памяти текущего состояния, что гарантировало защищенность данных от возможных повреждений и за счет ее редактирования обеспечивался возврат на предыдущие шаги обработки. Кроме того, в каждом состоянии модели предусматривалась обработка ошибок, возникающих в процессе обработки. С ростом потребностей в резервировании внешних гетерогенных источников требуется новое оснащение, устраняющее пробелы в реализации резервного копирования СОБД. Такого рода оснащение ранее не было предложено, в работе обсуждается его реализация, используется прототип программного обеспечения СОБД, сопровождающего процесс курсового проектирования по дисциплине «Базы данных».

***Ключевые слова*:** ситуационно-ориентированная база данных, встроенная динамическая модель, гетерогенные источники документов, резервное копирование, виртуальные мультидокументы, динамические объекты обработки данных, RESTful-сервисы.

Благодарности: работа выполнена при поддержке гранта РФФИ (грант 19-07-00682); результаты исследования, отражающие структуру разрабатываемого программного решения, были получены в рамках государственного задания № ДВУЭ-2020-0007.

***Для цитирования*:** Гусаренко А.С. Ситуационно-ориентированные базы данных: резервное копирование виртуальных мультидокументов модели динамических объектов обработки данных. *Моделирование, оптимизация и информационные технологии.* 2023;11(2). URL: https://moitvivt.ru/ru/journal/pdf?id=1322 DOI: 10.26102/2310-6018/2023.41.2.002 (на англ.).

Introduction

A priority task for any information system that manages data is backup or, in other words, the creation of "backups". Data about a business process, users, documents are valuable for any organization [1]. Possible incidents associated with their loss cannot be calculated in advance. In this regard, it is possible to insure against their loss by backing up to a drive separate from the infrastructure of the information system. Such a drive should allow accumulating backup copies of data, support several versions made at different times in order to capture the changes made in the information system at any time interval. The usual procedure is to store at least three backup versions separated by a time interval. The receiver can be a local hard drive without an Internet connection, an external flash drive, as well as a network drive or cloud storage. This applies to modern information systems, big data processing and the "Internet of things" [2].

Let us consider several levels for a backup task where data is stored. Backing up operating system data, where the settings of the operating system, its files, registry, settings are backed up. There are software solutions, data backup systems that helps to back up the entire user's personal computer. Other solutions are focused on maintaining backup copies of virtual machines running a server operating system that solves user tasks within the "client-server"

architecture. There are also built-in backup capabilities such as those in database management systems, they already have tools for creating backups in the form of detached backup files of only a separate type of database provision. Such files are called "data dumps". Subsequently, the dump file is opened in the restore tool on the database server and the restore process starts. It is impossible to simply copy the database file since it is opened by the DBMS process or locked by the DBMS server services; therefore, a specialized backup mechanism is provided that helps to detach the database or immediately create a backup using built-in tools. This mechanism creates backups and restores them in all modern relational DBMS, and the result of the restoration is no different from the original data subjected to backup.

The storage location for backups is usually a safe place, a local or network drive called a NAS, as well as cloud storage [3]. Nowadays, it is cloud technologies that are being promoted as a means of backup following cloud information systems. Cloud disks are used as a drive, copying to which is performed according to a schedule compiled by the user. Backups can be both virtual machine images and individual files, for example, the same database dumps. A specialized cloud for backups is not necessary, a regular cloud file storage with an accessible API [4] is also suitable for this task: a user application, operating system or DBMS can register in it and upload files in a protected mode on a schedule backups at the time specified by the administrator. The backup object is only required to implement automated functions that create backups, as well as an application that connects to the cloud storage to download files through the network. Created backups are managed by a single system administrator using cloud storage controls where it is possible to evaluate disk size, backup size, control backup versions, deleting outdated copies in a timely manner, and also to check their quality by test recovery.

This paper addresses a similar problem taking into account the specifics of designing situationally oriented databases (SODB). The virtual multi-document model underlying the SODB already at the current stage of development interacts with web services and data storages [5]. Based on this, it is possible to take a step towards managing cloud disks. The scientific and technical groundwork in this field is evident; it would help to achieve modern cloud backup using the cURL extension, which is widely used to work with external heterogeneous data sources of applications based on SODB [6, 7]. In situationally oriented databases, there are tools for virtual multi-documents / data arrays as well as dynamic data processing objects. By using them, heterogeneous data and methods for their processing are specified in the model. There may be a need to reserve such objects when the application is in operation. The known methods [8, 9] are not implemented in SODB and cannot be applied with the same efficiency. Manual methods for backing up heterogeneous SODB data are also losing their relevance due to the high labor intensity of performing routine operations. It should be noted that there are ways to solve the backup problem in SODB by developing a backup model, new elements and by creating an algorithm for processing a backup model or algorithms for uploading backups to cloud storage. These types of software are not available in SODB [10], but there is a potential in their development with a view to obtaining a qualitative result in the process of designing databases and creating software equipped with such capabilities.

Literature review

The problem of backup is well-known. Considering it in this paper, we reviewed and analyzed the available literature, articles, theses which have been published recently [11-13]. These sources are important because they deal with the management of new cloud technologies as well as new directions in the creation of modern non-relational NoSQL databases. One of these papers discusses the practical aspect of data analysis under the conditions of server operation limitations by creating backup copies for the analysis of out-of-order data from RAM. A program algorithm is being developed to determine the data to be deleted and those that

should be left for restoration in the storage. This type of manipulation is carried out without deploying a database server [14]. Papers were reviewed in order to find backup methods and tools available for transfer to SODB and algorithms that have proven their effectiveness in the direction of NoSQL. They are highly specialized, algorithms and software operate only with a certain type of data store, for example, key-value store backups.

Unfortunately, it is not possible to find easily portable backup methods that fully take into account the models and methods for processing heterogeneous data in the hierarchical situational model of SODB. Additional questions about the applicability of known methods are raised by the solutions proposed in the papers [15-17] if they are built from the standpoint of the polyglot persistence approach [18], based on the microservice architecture [19, 20]. There are recommendations for selecting backup systems. Their advantages and disadvantages are known [21, 22].

This article demonstrates how to model efficiently the use of the existing scientific and technical groundwork to solve the backup problem from the standpoint of a situation-oriented approach involving microservice architecture. There is currently no such support in the SODB; therefore, it is required to develop it in terms of a hierarchical situational model, backup methods in cloud storage, algorithmic support and software. The necessary conditions for this were created at the previous stages of research, where the problems of streaming data processing [23], extracting data from drawings of office documents [24] as well as working with heterogeneous sources in the form of relational databases were considered [25]. The necessary conditions for this were created at the previous stages of research that addressed the problems of streaming data processing [23], extracting data from drawings of office documents [24], as well as managing heterogeneous sources in the form of relational databases [27, 28].

Under the current conditions, it is not possible to find in published papers known methods for backing up SODB with a focus on the data processing model. There are no built-in tools for backing up virtual data arrays and data processing objects. In terms of algorithmic support, the problems of developing algorithms for the SODB backup model were not considered, so this paper focuses on their development in accordance with the backup model.

Backup of heterogeneous data sources and SODB virtual data processing objects

Reservation of heterogeneous data sources of the SODB architecture. The SODB architecture in the current conditions involves a large number of variants of heterogeneous data sources accepted for processing in the states of a hierarchical situational model. With the advent of SODB [26,28], these were the simplest XML documents stored on an external data carrier, web server hard drives. At that time, the system sources were not considered as heterogeneous. At this stage, the architecture has expanded significantly due to the involvement of various sources. If we turn to the modern architecture of the SODB in Figure 1, it shows how heterogeneous sources are represented not only from XML documents, but also from virtual data arrays, database servers, archives, microservices, third party web services. Sources, as they were added for processing in the SODB, seemed to be diverse in their internal representation and specialized tools were required for their processing. Such sources were characterized as heterogeneous, their distinctive features were presentation formats, access methods and processing methods [29, 30]. The existing SODB architecture has not fit into the traditional methods of solving the backup problem precisely because of the variety of access methods and the processing of internal content. Copying files manually to local media here no longer satisfies the requirements of the SODB infrastructure.

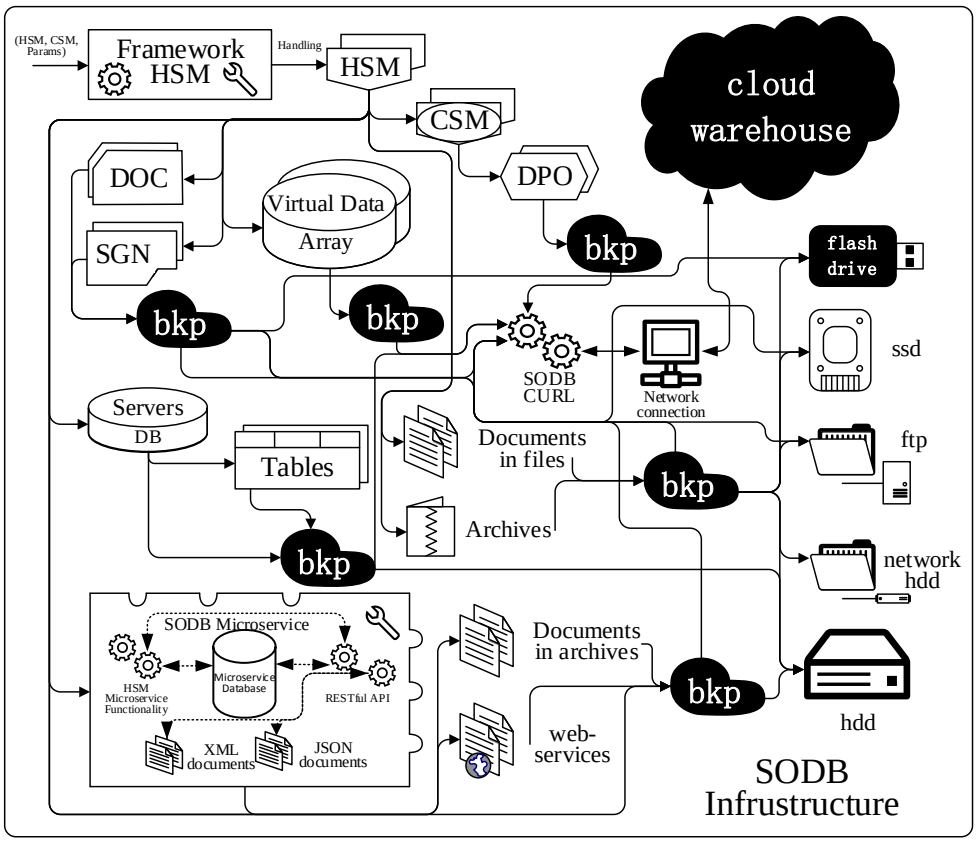

Figure 1 – SODB architecture with built-in elements of the backup model for heterogeneous data sources, virtual data processing objects

Рисунок 1 – Архитектура СОБД со встроенными элементами модели резервного копирования гетерогенных источников данных, виртуальных объектов обработки данных

If we imagine the backup process manually, then this means that operating an automated mode requires to access each heterogeneous source and, taking into consideration the writing of query scripts for each source, copy data to any media. This method does not account for the intermediate and final resulting representations of data sources, data retrieved during the operation of the SODB. In addition, there is a problem with the further growth of connecting new heterogeneous sources and the increase in the complexity of their reservation.

Possibilities of backing up heterogeneous SODB sources to cloud data storage. With the development of modern backup technologies, traditional storage media such as HDDs, SSDs and Flash-drives are also being used. Since SODB runs on a web server, FTP servers and network drives are available for use to place their backups there. It can be achieved by connecting to the data storage server. Recently, cloud backup technology has become popular, in Figure 1 it is depicted as "Cloud Warehouse". Such a storage is a user universal tool for storing information in the form of files with convenient functionality, controls and buttons in a web browser. For technical specialists who can use this cloud for their own purposes, documentation [4] is available which describes in detail how to connect to the cloud programmatically and start creating directories in it, uploading files to them and vice versa. Access to the cloud is granted by registering an application in a storage account, which issues a token that can be programmatically accessed at the login address. In this case, the SODB is represented as an application registered in the cloud storage, sending requests after authorization to create directories and requests that help to upload files to these directories using the PUT method and the HTTPS protocol. Previously, detailed studies were carried out on SODB [5] related to the implementation of cURL tools for managing digital web services.

SODB microservices were created now acting as their own heterogeneous data. Such sources in Figure 1 also need to be backed up under current conditions, almost all of them somehow work using the HTTPS protocol and data from them is requested via the Internet. For each type of source from Figure 1, specialized `bkp` elements are introduced to solve the backup task; due to their specification, it is indicated which source will be used and which method will be used to initiate the backup procedure as well as to which data store. The `bkp` elements were not previously introduced, they have been introduced to solve the problem of reserving specific heterogeneous sources because for each source, dynamic data processing object, upon the request of a system analyst or system administrator, the need to create backups is determined,

Virtual multi-document model with a backup element introduced into it. From an architectural point of view, Figure 1 demonstrates the proposed approach to the backup task in SODB. Here, this proposal based on the hierarchical situational model of the SODB, which entails the use of the model as a backup management tool, develops further. This approach is illustrated in the example from Figure 2 where the `bkp` backup element, like all elements of this type, is connected directly to a heterogeneous source to solve the redundancy problem. These new elements may contain parameters detailing the backup procedure and methods invoked to create the copy or may refer to a virtual multidocument model where operations and access key parameters are specified and structured like a normal external RESTful web service. The model of such style does not undergo significant changes, and based on the already familiar multi-document model, you can implement the entire backup as indicated by the example to the cloud. At the same time, the model does not change significantly, so this solution is very close to the previously introduced invariance principle [29], where changing the type of a heterogeneous source does not significantly affect the model and it remains unchanged.

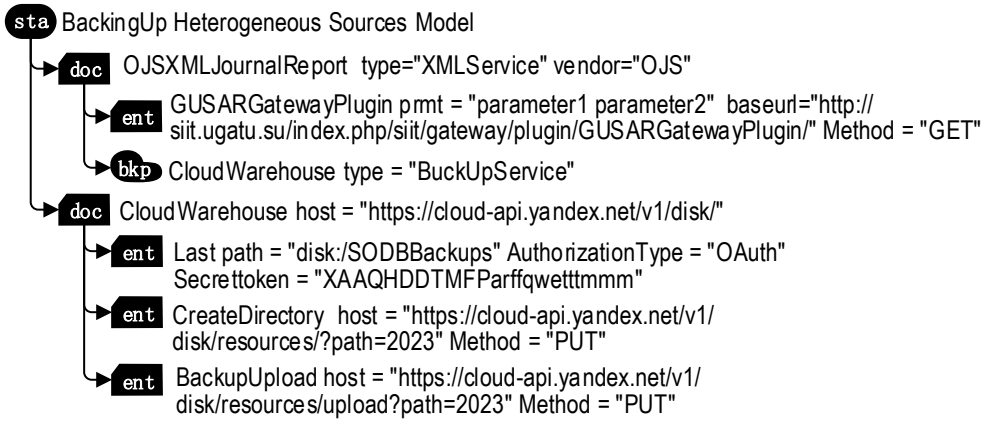

Figure 2 – Fragment of the hierarchical situational model of the SODB with the bkp element introduced into the heterogeneous data source and the specified virtual document of the data warehouse providing backup

Рисунок 2 – Фрагмент иерархической ситуационной модели СОБД с введенным в гетерогенный источник данных элементом bkp и заданным виртуальным документом хранилища данных, обеспечивающим резервное копирование

Figure 2 demonstrates an image of a hierarchical situational model fragment, where the state for backup is set. Within this state, two virtual multi-documents are specified. The first virtual multi-document `OJSXMLJournalReport` is a heterogeneous document of the XML service type based on the journal system of the Open Journal Systems platform. The virtual document is structured by an entry element nested in it with the parameters specified in it, the entry address and the GET method for requesting information via the HTTP protocol are provided in the XML file. The parameters in this case regulate the depth of data coverage, the

features of the desired result, similar to an SQL query, but without specifying special technical details. Next is the nested `bkp`-element introduced in this paper, which has a backup type, the name `CloudWarehouse`. This name matches the name of the next virtual multi-document, which describes exactly the virtual storage where the backup copy of the document received from the `OJSXMLJournalReport` service is stored.

The second virtual multi-document is structured by entry-elements that have their own purpose.

1. The `Last` entry element provides authorization; each request to the cloud storage requires OAuth authorization with a secret token obtained when registering the SODB application in the cloud storage;

2. The `CreateDirectory` entry element ensures the creation of the `SODBBackups` directory in the cloud storage for placing heterogeneous source backups into it using the PUT method;

3. Entry-element `BackupUpload` provides direct upload of the backup file to the cloud in the directory specified in the previous step.

As can be seen from Figure 2, the proposed backup solution fits into the concept of a virtual multi-document of the SODB model. A new element is introduced and the existing elements of virtual multi-documents support these innovations. Thus, the solution of the backup problem is provided by means of the SODB model and completely based on it, and on the other hand, the previously used cURL methods also allow solving this problem at the level of situational modeling and enable backup management at a high level of abstraction.

Setting specialized elements of the SODB model for the problem of backing up heterogeneous sources and data processing objects

The bkp elements inside dynamic data processing objects. The `bkp` elements proposed in this paper can be contained not only inside heterogeneous sources, but also inside dynamic data processing objects. The binding of the `bkp` element helps to reserve not the initial state of the source, in which it is presented in some service or hard disk, but, for example, when it is processed, that is, the intermediate state between the initial and the resulting state is reserved. Dynamic data processing objects occupy quite massive volumes due to their content, actions performed, specifications for filtering or merging data. Under these conditions, it is possible to specify a `bkp` element with minimal reservation details, but such an element will refer to a structured virtual multidocument, where there will already be a more detailed structuring of the operations performed. Figure 3 shows a fragment of such a model, where the target dynamic data processing object `DPO_Load_Data` contains a link in the `src` element to the multi-document `Report` obliging to accept for processing 1000 nodes from the SODB microservice in JSON format. Inside the `DPO_Load_Data` object, there is a `bkp` element named `CloudWarehouse` that refers to a multi-document with the details of starting the backup process and accessing the cloud storage of documents. Intermediate states of the `DPO_Load_Data` object are reserved, so `act` elements are usually indicated inside it, leading the document to intermediate stages due to the implementation of CRUD operations. The `act` elements are usually automated functions, operations that are nested in groups in a model, submodel or dynamic data processing object. Here and below, the examples do not indicate the features of processing and the use of functions specified in the action elements of the model.

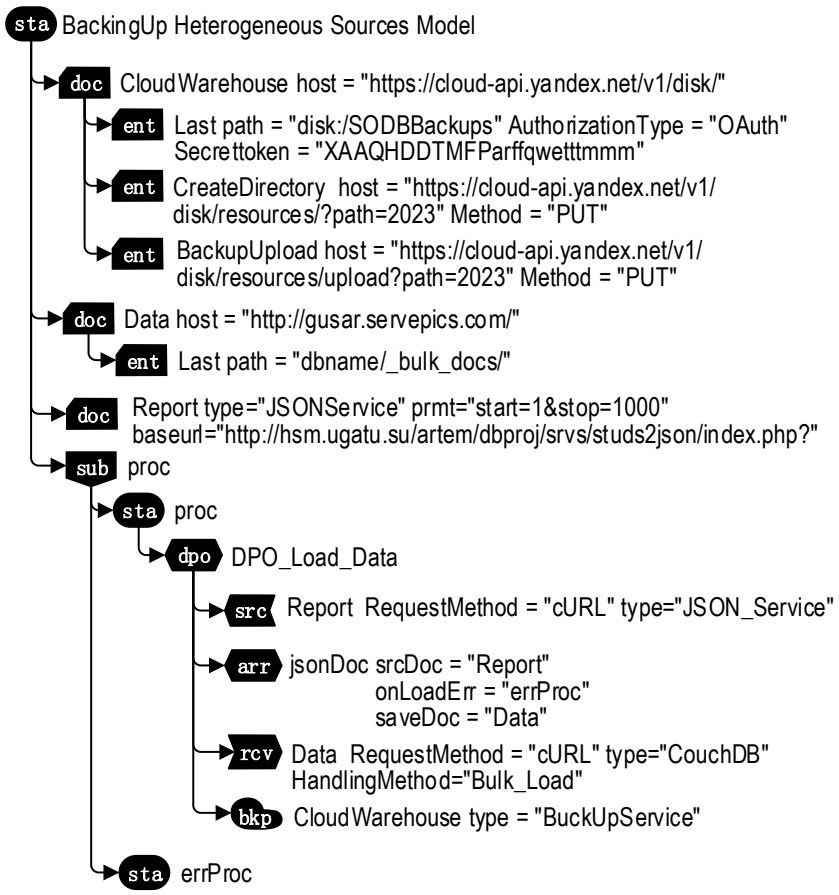

Figure 3 – Fragment of the hierarchical situational model of the SODB with the bkp element introduced into the virtual data processing object and the specified virtual data storage document providing backup

Рисунок 3 – Фрагмент иерархической ситуационной модели СОБД с введенным в виртуальный объект обработки данных элементом bkp и заданным виртуальным документом хранилища данных, обеспечивающим резервное копирование

The entire backup process is described in the `CloudWarehouse` virtual multi-document and consists of four steps:

1) setting the access point `https://cloud-api.yandex.net/v1/disk/` to cloud storage;

2) authorization in cloud storage through the use of the `OAuth` mechanism with a secret token `Secrettoken`;

3) creating the `disk:/SODBBackups` directory in the cloud storage using the PUT method;

4) uploading a backup to the cloud storage in the created directory using the PUT method.

As a result of initiating the backup procedure in the cloud storage, backup copies of dynamic data processing objects with intermediate stages of data processing are saved in the form of files. A standard cloud for storing files from one of the suppliers of well-known Russian search engines has chosen as cloud storage.

Map-Reduce View Document Backup for Big Data Store from Heterogeneous Processing Dynamic Objects. Similarly, a data processing object that creates views in the cloud storage for processing large documents can be reserved. Figure 4 shows a fragment of the reservation of the representation created in the big data processing warehouse, while it should

be noted that not only the functions themselves contained in the SODB functions catalog are reserved, but their synthesized version obtained in the data processing object, so this is a ready-made document defined in in terms of big data storage as a view [32].

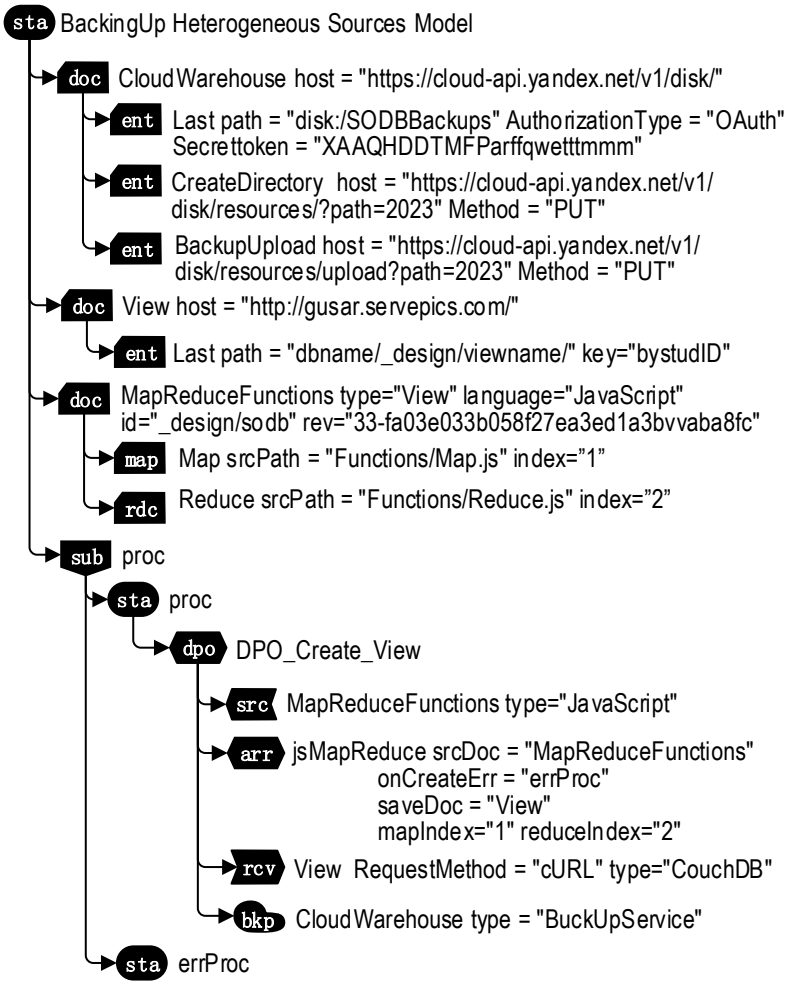

Figure 4 – Fragment of the hierarchical situational model of the SODB with the bkp element for backing up the Map and Reduce functions of the external data store inside the dynamic data processing object

Рисунок 4 – Фрагмент иерархической ситуационной модели СОБД с элементом bkp для резервного копирования функций Map и Reduce внешнего хранилища данных внутри динамического объекта обработки данных

The example gives only two functions Map and Reduce from the library, but there were examples when several of these functions were most often used, thereby making it possible to create different views with combinations of JavaScript functions. Thus, different representations are backed up during processing.

Backup of documents verified in cryptographic services. Documents that have legal significance and are verified [33] in cryptographic services can also be subject to reservation. They are stored in `doc` elements denoting a multi-document with given detached electronic signatures in nested `sgn` elements. Figure 5 shows a fragment of the electronic document verification model in a cryptographic service, but with the `bkp` element embedded in the multi-document to help ensure backup to the cloud storage.

Processing a fragment of a model of a heterogeneous data source with a given bkp element. To process a model with built-in `bkp` elements, an algorithm that briefly describes not

only the backup method itself, but also the processing of model elements, parameters is needed. It is also required to ensure the transition from the backup element of a heterogeneous source to the processing of multi-document specifications containing the parameters for connecting to the cloud storage, the transfer of the document for downloading with a call to the connection method to perform CRUD operations inside the cloud. This algorithm has been developed and proposed for processing the backup model of a heterogeneous data source, while the `bkp` element is not specified in the dynamic data processing object. The element is specified in the heterogeneous source itself, and its state is backed up in the cloud without intermediate stages, that is, only the initial state and the resulting.

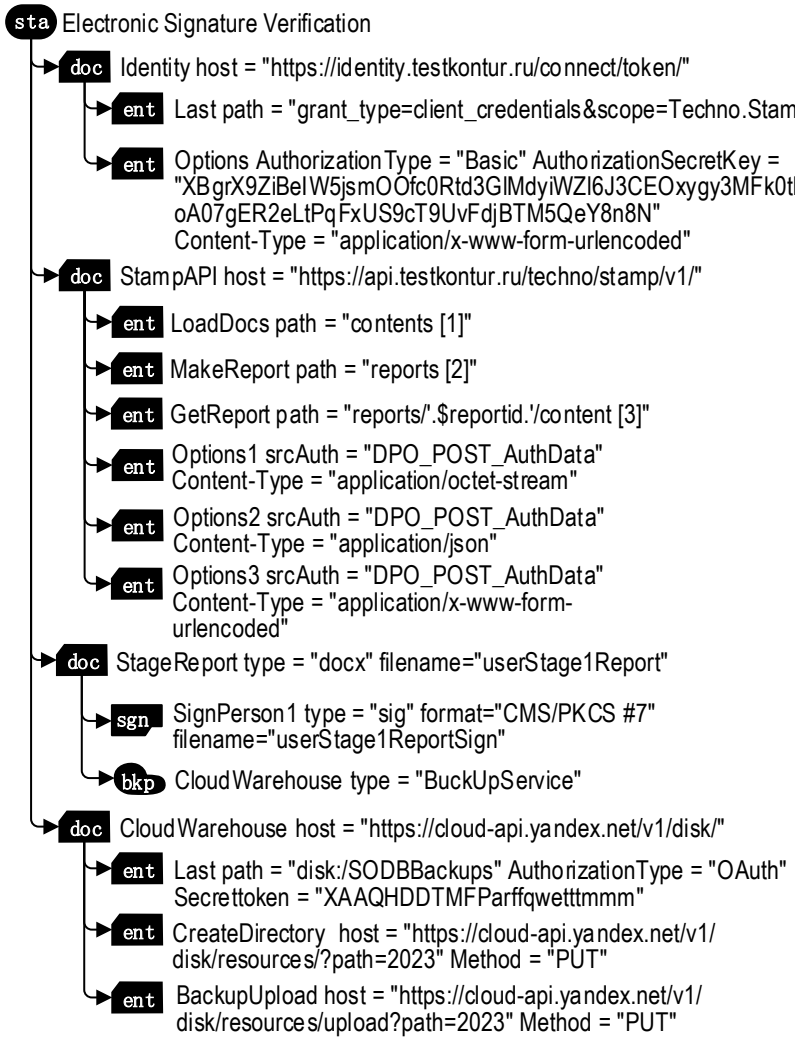

Figure 5 – Fragment of a hierarchical situational model of SODB with an example of specifying the bkp element to ensure document backup along with an electronic signature

Рисунок 5 – Фрагмент иерархической ситуационной модели СОБД с примером задания элемента bkp для обеспечения резервного копирования документа вместе с электронной подписью

Figure 6 shows this algorithm in pseudocode in an accessible form explaining the redundancy process for one heterogeneous source.

```
000 Begin
001   sta_multidoc = New State ('BackingUp')
002   ForEach sta_multidoc.doc: doc.Name
003      if exists element (ent)
004         ForEach ent: e.Name, e.BaseUrl, e.Parameters, e.Method do
005            e.Name = add.NameEnt, e.BaseUrl = add.addBaseUrl,
006            e.Parameters = add.Parameters, e.Method = add.Method
007      if exists element (bkp)
008         ForEach bkp: bkp.Name, bkp.Type do
009            bkp.Name = add.Name, bkp.Type = add.Type
010            for (doc.Name)
011               d = e.Method(e.BaseUrl, e.Parameters)
012      if exists element (doc.Name==doc.bkp.Name)
013         b = doc.Name
014         if exists element (b.ent)
015            ForEach b.ent: b.Host, b.Path, b.AuthorizationType,
016            b.SecretToken, b.Method do
017               b.Host = add.Host, b.Path = add.Path,
018               b.AuthorizationType=add.AuthorizationType,
019               b.SecretToken=add.SecretToken, b.Method=add.Method
020      m = b.Method(d, b.Host, b.Path, b.AuthorizationType, b.SecretToken)
021      return m
022 End
```

Figure 6 – Pseudo-code of the algorithm for processing a fragment of a heterogeneous data source model with a given bkp element that provides data backup into cloud storage

Рисунок 6 – Псевдокод алгоритма обработки фрагмента модели гетерогенного источника данных с заданным элементом bkp, обеспечивающим резервное копирование данных в облачное хранилище

Similar algorithms can also be developed for any other heterogeneous sources, dynamic data processing objects, and in this case, we will use an example of such an algorithm and analyze the most important stages of backup to cloud storage. If we consider the algorithm line by line, we can distinguish the following stages of the backup process.

1. Line 000–001 initiates the algorithm, a new multidocument model is created.

2. Line 002–001 enumerates all `doc` elements of the multidocument model by their names.

3. Line 003–006 if there are `entry`-elements inside the multi-document, then parameters are collected for each `entry`-element that enable further processing.

4. Line 007–011 if there are `bkp` elements inside the multi-document, then for each `bkp` element, parameters that enable backup to cloud storage are collected.

5. Line 010–011 for a multidocument with the given name, a method is called that requests this from a heterogeneous source, data from the source is retrieved.

6. Line 012–013 transition from the backup element to the multidocument of the same name with specifications for access to the cloud storage, the name of the multidocument is saved.

7. Line 014–019 nested entry-elements of this multi-document of the same name are processed; if they exist, their parameters are collected for each entry-element.

8. Line 020–021 is the final stage of the algorithm, where for the multidocument of the same name with the name that matches the nested `bkp` element, the method of loading the created backup copy is applied. The method receives the document contained in the variable d and the parameters of access to the cloud storage, including the save path in the cloud.

Practical implementation of backup based on the backup model in SODB

A practical example of a backup model in a synthesized form is shown in Figure 7. At the current stages of development, the SODB includes data storages. At one of the stages, the management of document-oriented storage is organized [32], then the storage of a cryptographic service; next, cloud storage is connected. From experience, it can be deduced that in modern conditions a connection between SODB with some kind of data warehouse is regularly formed. In theory, a data warehouse is understood as a subject-oriented system where information system operation data related to its operational activities are accumulated. The data warehouse has a specific structure that provides high data processing performance to solve the problem of creating business intelligence reports in order to support decision making. In this sense, the SODB needs further development under the decision-making process, the creation of business reports to support decision-making. This example uses file cloud storage for backup purposes.

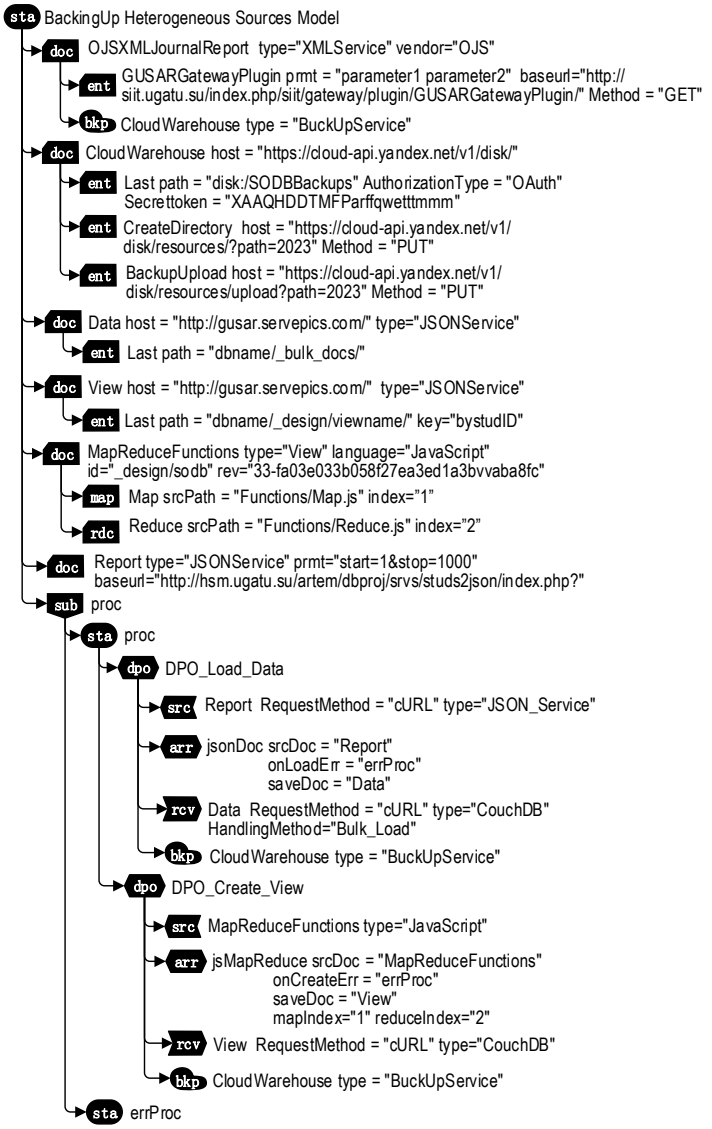

Figure 7 – General view of the hierarchical situational model of SODB for backing up heterogeneous data sources and dynamic data processing objects to cloud storage

Рисунок 7 – Общий вид иерархической ситуационной модели СОБД для резервного копирования гетерогенных источников данных и динамических объектов обработки данных в облачное хранилище

Conceptual transition from SODB to situationally oriented data warehouse (SODW). Situation-oriented databases now use a document-oriented data store, operate on data in them, provide the execution of Map and Reduce statistical functions in order to process large documents. Thus, analytical capabilities are employed, which is inherent in the functionality of data warehouses. With the expansion of SODB capabilities, the addition of microservices to the SODB architecture is preparing a transition at the concept level to a situationally oriented data warehouse. Situation-oriented data warehouse (SODW) expands the capabilities of SODB. It is marked not only by its ability to process data, but also to manage multiple data warehouses and create business intelligence reports. SODB is a document-oriented database, but it already uses servers of relational and non-relational heterogeneous data sources, so the current definition does not fully correspond to the full range of functionality of such a development. It should be noted here that this is a potential future direction in the research on hierarchical situational modeling. A situation-oriented data warehouse can also effectively manage both databases and documents, which would tie it to the non-relational direction of database development.

To organize the storage and processing of data, it must contain large amounts of data. Figures 7 and 8 show a database model from practice, on course design, which contains large amounts of information on course design in "Databases" course. Based on such data, it is possible to create reports with summary tables and charts on progress, completion of the design stages, typical errors and frequently occurring comments on the completed work.

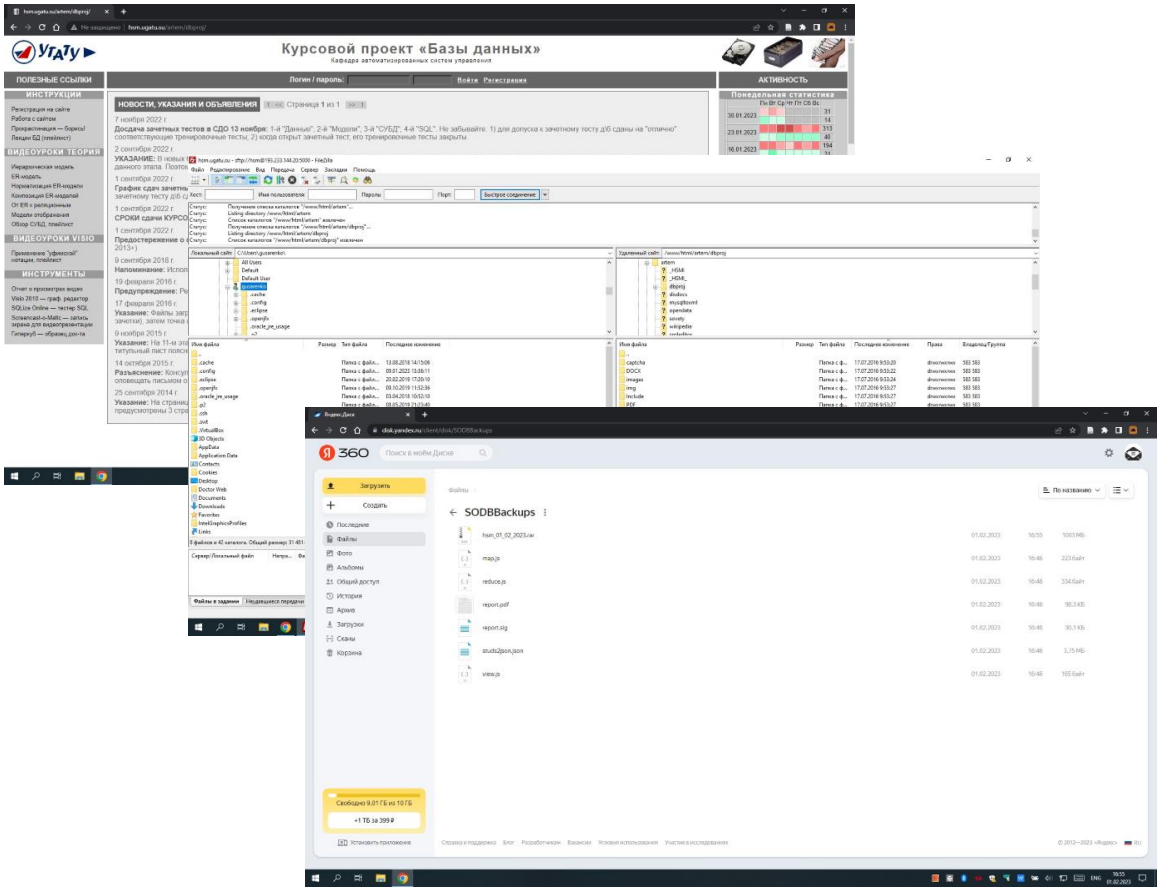

Figure 8 – Screen forms of the "Course Design" website based on SODB in "Databases" course, directories of the FTP-server of the SODB-project, cloud storage with uploaded backups

Рисунок 8 – Экранные формы веб-сайта «Курсового проектирования» на основе СОБД по дисциплине «Базы данных», каталоги FTP-сервера проекта СОБД, облачное хранилище с загруженными резервными копиями

In this article, at the current stage, a groundwork future has been created by providing research on cloud data warehouses where backup copies of heterogeneous sources are placed. As a result of processing the model from Figure 7, backup copies of heterogeneous SODB sources should be stored in the cloud using the algorithm and plug-in methods. Figure 8 shows the screen forms of the SODB project with the main page of the SODB course design, files located in the local directory of the server in relation to external heterogeneous sources. The third screen form demonstrates the backups uploaded to the cloud storage in the form of backup files of heterogeneous SODB sources. The implementation of methods and specific features of managing a certain cloud data storage in this case is the responsibility of the programmer who embeds the backup modules in the SODB interpreter. On the basis of this software which has the ability to interact with the cloud data storage, research on the organization of situationally oriented data storages can be continued.

Conclusion

The research is aimed at solving the relevant issue of backing up heterogeneous data sources in SODB. Previously, this task was not given due attention despite the development of backup technologies around the world. Here we used the best developments in SODB, together with network data exchange technologies as well as modern cloud storage. All suggestions in this article are based on the best database design practices as well as a situational approach to achieve the goals of a qualitative increase in performance in the design of databases and applications at a high level of abstraction reducing the complexity of the overall backup process. An architecture has been developed taking into account the backup process; backup models have been built with elements introduced into the model for various types of heterogeneous sources and dynamic data processing objects. An algorithm has been developed for processing a fragment of a model of a heterogeneous data source with a specified element that provides data backup to cloud storage. The example demonstrates the possibility of backing up a SODB project to one of the popular cloud data storages. As a research prototype for implementation, a course design project on databases is considered. The proposed specifications make it possible to make the model more compact, as well as to use the virtual document model to control the methods and modes of access to the cloud data storage. Also, the model with the introduction of new elements allows making minimal changes in the model to solve the problem of backup due to the observance of the principle of invariance.

REFERENCES

1. Shaytura S.V., Pitkevich P.N. Data backup methods for mission-critical information systems. *Rossiyskiy tekhnologicheskiy zhurnal = Russian Technological Journal*. 2022;10(1):28–34. DOI: 10.32362/2500-316X-2022-10-1-28-34. (In Russ.).
2. Chang D., Li L., Chang Y., Qiao Z. Cloud Computing Storage Backup and Recovery Strategy Based on Secure IoT and Spark. *Mobile Information Systems*. 2021:9505249. DOI: 10.1155/2021/9505249.
3. Mustafin A. Transferring data to the clouds: how to do it and how to protect yourself. *Sistemnyy administrator = System Administrator*. 2021;10(227):46–48. (In Russ.).
4. Introduction – API. Yandex.Disk API. Available from: https://yandex.ru/dev/disk/api/concepts/about.html?userAgreement=%2Flegal%2Fdisk_api%2F (accessed on 25.01.2023). (In Russ.).
5. Mironov V.V., Gusarenko A.S., Yusupova N.I. Monitoring YouTube video views in the educational environment based on situation-oriented database and RESTful Web Services. *Sistemnaya inzheneriya i informatsionnye tekhnologii = Systems Engineering and Information Technologies*. 2021;3(1(5)):39–49.

6. Gusarenko A.S. Certificate of state registration of the computer program No. 2022617505. Situational database modules for extracting large documents and archives from RESTful services of heterogeneous data stores. 2022. (In Russ.).
7. Wilde E., Pautasso C. *REST: From Research to Practice*. Springer Science & Business Media; 2011. 528 p. DOI: 10.1007/978-1-4419-8303-9.
8. Taranin S.M. Backup with Storage in a Database. *Modelirovaniye i analiz informatsionnykh sistem = Modeling and Analysis of Information Systems*. 2016;23(4):479–491. DOI: 10.18255/1818-1015-2016-4-479-491 (In Russ.).
9. Chakravarthy S.R., Shruti, Kulshrestha R. A queueing model with server breakdowns, repairs, vacations, and backup server. *Operations Research Perpsectives*. 2020;7:100131. DOI: 10.1016/j.orp.2019.100131
10. Mironov V.V., Gusarenko A.S., Yusupova N.I. Situation-oriented databases: processing office documents. *Modeling, Optimization and Information Technology*. 2022;10(2). Available from: https://moitvivt.ru/ru/journal/pdf?id=1187 DOI: 10.26102/2310-6018/2022.37.2.021 (accessed on 25.11.2022).
11. Kishenya P. How to organize backup: analysis of errors, strategies and recommendations. *Sistemnyy administrator = System Administrator*. 2020;3(208):10–12. (In Russ.).
12. Kalinin M. How justified is backup data storage in the cloud? *Sistemnyy administrator = System administrator*. 2020;3(208):1–13. (In Russ.).
13. Laskov D., Frolov S. Backup, recovery and replication: choosing a backup system for an average company. *Sistemnyy administrator = System Administrator*. 2020;3(208):14–17. (In Russ.).
14. Tikhonov N.A., Budnikova I.K. Redis backup analysis and processing. *Informacionnye tehnologii v stroitel'nyh, social'nyh i jekonomicheskih sistemah*. 2020;2(20):121–124. (In Russ.).
15. Bartsevich A.V., Polyakov V.V., Shaikamalov M.R. Algorithm for selecting a backup scheme for a complex technical system. *Uspekhi sovremennoy radioelektroniki = Achievements of Modern Radioelectronics*. 2020;74(8):34–39. DOI: 10.18127/j20700784-202008-03 (In Russ.).
16. Savin I.V. Features of ensuring the reliability of stability, preservation and availability of data. *Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskiye nauki = Izvestiya Tula State University. Technical science.* 2019;3:118–122. (In Russ.).
17. Patutina A.M., Rudakov I.V. Determination of data backup system fault tolerance. *Vestnik moskovskogo gosudarstvennogo tekhnicheskogo universiteta Im. N. Baumana. Seriya Priborostroyeniye = Herald of the Bauman Moscow State Technical University. Series Instrument Engineering*. 2019;5(128):64–78. DOI: 10.18698/0236-3933-2019-5-64-78 (In Russ.).
18. Kolonko M., Mullenbach S., Polyglot Persistence in conceptual modeling for information analysis. *In ACIT'2020: Proc. 10th Int. Conf. on Advanced Computer Information Technologies*. 2020:590–594. DOI: 10.1109/ACIT49673.2020.9208928.
19. Huf A., Siqueira F. Composition of heterogeneous web services: A systematic review. *Journal of Network and Computer Applications*. 2019;143:89–110. DOI: 10.1016/j.jnca.2019.06.008.
20. Hamzehloui M.S., Sahibuddin S., Salah K. A systematic mapping study on microservices. *Advances in Intelligent Systems and Computing*. 2019;843:1079–90. DOI: 10.1007/978-3-319-99007-1_100.
21. Bopp V.A. Backup technology. advantages and disadvantages. *Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskiye nauki = Izvestiya Tula State University. Technical science.* 2019;3:134–137. (In Russ.).
22. Bopp V.A. Features of the choice of backup systems. *Izvestiya Tul'skogo*

gosudarstvennogo universiteta. Tekhnicheskiye nauki = Izvestiya Tula State University. Technical science. 2019;10:297–300. (In Russ.).
23. Gusarenko A.S., Mironov V.V., Yusupova N.I. Stream processing of large documents in situationally oriented databases. *Trudy 6-oy mezhdunarodnoy konferentsii po Informatsionnym tekhnologiyami intellektual'noy podderzhki prinyatiya resheniy = Proceedings of the 6-th International Conference Information Technologies for Intelligent Decision Making Support*. Ufa, USATU; 2018. P. 7–12. (In Russ.).
24. Mironov V.V., Gusarenko A.S., Tuguzbaev G.A. Extracting semantic information from graphic schemes. *Informatika i avtomatizatsiya = Informatics and Automation*. 2021;20(4):940–970. DOI: 10.15622/IA.20.4.7. (In Russ.).
25. Gusarenko A.S. Improvement of situation-oriented database model for interaction with Mysql. *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie = Journal of Instrument Engineering*. 2016;59(5):355–363. DOI: 10.17586/0021-3454-2016-59-5-355-363. (In Russ.).
26. Mironov V.V., Gusarenko A.S., Yusupova N.I. Situation-oriented databases: polyglot persistence based on REST microservices. *Prikladnaya informatika = Applied Informatics.* 2019;14(5):87–97. DOI: 10.24411/1993-8314-2019-10038. (In Russ.).
27. Mironov V., Gusarenko A., Yusupova N. Stream documents processing invariance in situation-oriented databases. *In: 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS'2019)*. Atlantis Press; 2019:309–315. DOI: 10.2991/itids-19.2019.55.
28. Mironov V.V., Gusarenko A.S., Yusupova N.I. Building of virtual multidocuments mapping to real sources of data in situation-oriented databases. *Communications in Computer and Information Science*. 2021;1204:167–178. DOI: 10.1007/978-3-030-78273-3_17.
29. Mironov V.V., Gusarenko A.S., Yusupova N.I. Structuring Virtual Multi-Documents in Situationally-Oriented Databases by Means of Entry-Elements. *Informatika i avtomatizatsiya (Trudy SPIIRAN) = Informatics and Automation (SPIIRAS Proceedings)*. 2017;53(4):225–243. DOI: 10.15622/sp.53.11. (In Russ.).
30. Mironov V.V., Gusarenko A.S., Yusupova N.I. The Invariance of the Virtual Data in The Situationally Oriented Database When Displayed on Heterogeneous Data Storages. *Vestnik komp'iuternykh i informatsionnykh tekhnologii = Herald of Computer and Information Technologies*. 2017;1(151):29–36. DOI: 10.14489/VKIT.2017.01.PP.029-036. (In Russ.).
31. "Databases" course project. Available from: http://hsm.ugatu.su/artem/dbproj/ (accessed on 20.10.2022) (In Russ.).
32. Gusarenko A.S. Situation-oriented databases: processing heterogeneous documents of microservices in a document-based storage. *Modeling, Optimization and Information Technology*. 2022;10(4). Available from: https://moitvivt.ru/ru/journal/pdf?id=1247. DOI: 10.26102/2310-6018/2022.39.4.003 (accessed on 25.01.2023).
33. Gusarenko A.S. Situation-oriented databases: verifying electronic signatures of heterogeneous documents in a RESTful web service. *Modeling, Optimization and Information Technology.* 2022;10(4). Available from: https://moitvivt.ru/ru/journal/pdf?id=1281. DOI: 10.26102/2310-6018/2022.39.4.019 (accessed on 25.01.2023).

СПИСОК ИСТОЧНИКОВ

1. Шайтура С.В., Питкевич П.Н. Методы резервирования данных для критически важных информационных систем предприятия. *Российский технологический журнал*. 2022;10(1):28−34. DOI: 10.32362/2500-316X-2022-10-1-28-34.

2. Chang D., Li L., Chang Y., Qiao Z. Cloud Computing Storage Backup and Recovery Strategy Based on Secure IoT and Spark. *Mobile Information Systems*. 2021:9505249. DOI: 10.1155/2021/9505249.
3. Мустафин А. Перенос данных в облака: как сделать и как защититься. *Системный администратор*. 2021;10(227):46–48.
4. Введение – API. API Яндекс.Диска. Доступно по: https://yandex.ru/dev/disk/api/concepts/about.html?userAgreement=%2Flegal%2Fdisk_api%2F (дата обращения: 25.01.2023 г.).
5. Mironov V.V., Gusarenko A.S., Yusupova N.I. Monitoring YouTube video views in the educational environment based on situation-oriented database and RESTful Web Services. *Sistemnaya inzheneriya i informatsionnye tekhnologii = Systems Engineering and Information Technologies*. 2021;3(1(5)):39–49.
6. Гусаренко А.С. Свидетельство о государственной регистрации программы для ЭВМ № 2022617505. Модули ситуационно-ориентированной базы данных для извлечения больших документов и архивов из RESTful-сервисов гетерогенных хранилищ данных. 2022.
7. Wilde E., Pautasso C. *REST: From Research to Practice*. Springer Science & Business Media. 2011. 528 p. DOI: 10.1007/978-1-4419-8303-9.
8. Таранин С.М. Резервное копирование с хранением в базе данных. *Моделирование и анализ информационных систем*. 2016;23(4):479–491. DOI: 10.18255/1818-1015-2016-4-479-491.
9. Chakravarthy S.R., Shruti, Kulshrestha R. A queueing model with server breakdowns, repairs, vacations, and backup server. *Operations Research Perpsectives*. 2020;7:100131. DOI: 10.1016/j.orp.2019.100131.
10. Mironov V.V., Gusarenko A.S., Yusupova N.I. Situation-oriented databases: processing office documents. *Modeling, Optimization and Information Technology*. 2022;10(2). Available from: https://moitvivt.ru/ru/journal/pdf?id=1187. DOI: 10.26102/2310-6018/2022.37.2.021 (дата обращения: 25.11.2022).
11. Кишеня П. Как организовать резервное копирование: разбор ошибок, стратегий и рекомендаций. *Системный администратор*. 2020;3(208):10–12.
12. Калинин М. Насколько оправдано резервное хранение данных в облаке? *Системный администратор*. 2020;3(208):1–13.
13. Ласьков Д., Фролов С. Backup, восстановление и репликация: выбираем систему резервного копирования для средней компании. *Системный администратор*. 2020;3(208):14–17.
14. Тихонов Н.А., Будникова И.К. Анализ и обработка резервных копий Redis. *Информационные технологии в строительных, социальных и экономических системах*. 2020;2(20):121–124.
15. Барцевич А.В., Поляков В.В., Шайкамалов М.Р. Алгоритм выбора схемы резервирования сложной технической системы. *Успехи современной радиоэлектроники*. 2020;74(8):34–39. DOI: 10.18127/j20700784-202008-03.
16. Савин И.В. Особенности обеспечения отказоустойчивости, сохранности и доступности данных. *Известия тульского государственного университета. Технические науки*. 2019;3:118–122.
17. Патутина А.М., Рудаков И.В. Определение отказоустойчивости системы резервного копирования данных. *Вестник московского государственного технического университета Им. Н. Баумана. Серия Приборостроение*. 2019;5(128):64–78. DOI: 10.18698/0236-3933-2019-5-64-78.
18. Kolonko M., Mullenbach S., Polyglot Persistence in conceptual modeling for information analysis. *In ACIT'2020: Proc. 10th Int. Conf. on Advanced Computer Information*

Technologies. 2020:590–594. DOI: 10.1109/ACIT49673.2020.9208928.
19. Huf A., Siqueira F. Composition of heterogeneous web services: A systematic review. *Journal of Network and Computer Applications*. 2019;143:89–110. DOI: 10.1016/j.jnca.2019.06.008.
20. Hamzehloui M.S., Sahibuddin S., Salah K. A systematic mapping study on microservices. *Advances in Intelligent Systems and Computing*. 2019;843:1079–90. DOI: 10.1007/978-3-319-99007-1_100.
21. Бопп В.А. Технология резервного копирования. Преимущества и недостатки. *Известия тульского государственного университета. Технические науки*. 2019;3:134–137.
22. Бопп В.А. Особенности выбора систем резервного копирования. *Известия Тульского Государственного Университета. Технические Науки*. 2019;10:297–300.
23. Гусаренко А.С., Миронов В.В., Юсупова Н.И. Потоковая обработка больших документов в ситуационно-ориентированных базах данных. *Труды 6-ой международной конференции по Информационным технологиями интеллектуальной поддержки принятия решений*. Уфа, Россия: УГАТУ; 2018. С. 7–12.
24. Миронов В.В., Гусаренко А.С., Тугузбаев Г.А. Извлечение семантической информации из графических схем. *Информатика и автоматизация*. 2021;20(4):940–970. DOI: 10.15622/IA.20.4.7.
25. Гусаренко А.С. Усовершенствование модели ситуационно-ориентированной базы данных для взаимодействия с MySQL. *Известия высших учебных заведений Приборостроение*. 2016;59(5):355–363. DOI: 10.17586/0021-3454-2016-59-5-355-363.
26. Миронов В.В., Гусаренко А.С., Юсупова Н.И. Ситуационно-ориентированные базы данных: polyglot persistence на основе REST-микросервисов. *Прикладная информатика*. 2019;14(5):87–97. DOI: 10.24411/1993-8314-2019-10038.
27. Mironov V., Gusarenko A., Yusupova N. Stream documents processing invariance in situation-oriented databases. *In: 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS'2019)*. Atlantis Press; 2019:309–315. DOI: 10.2991/itids-19.2019.55.
28. Mironov V.V., Gusarenko A.S., Yusupova N.I. Building of virtual multidocuments mapping to real sources of data in situation-oriented databases. *Communications in Computer and Information Science*. 2021;1204:167–178. DOI: 10.1007/978-3-030-78273-3_17.
29. Миронов В.В., Гусаренко А.С., Юсупова Н.И. Структурирование виртуальных мультидокументов в ситуационно-ориентированных базах данных с помощью entry-элементов. *Информатика и автоматизация (Труды СПИИРАН)*. 2017;53(4):225–243. DOI: 10.15622/sp.53.11.
30. Миронов В.В., Гусаренко А.С., Юсупова Н.И. Инвариантность виртуальных данных в ситуационно-ориентированной базе данных при отображении на разнородные хранилища. *Вестник компьютерных и информационных технологий*. 2017;1(151):29–36. DOI: 10.14489/VKIT.2017.01.PP.029-036.
31. Курсовой проект «Базы данных». Доступно по: http://hsm.ugatu.su/artem/dbproj/ (дата обращения: 20.10.2022).
32. Гусаренко А.С. Ситуационно-ориентированные базы данных: обработка гетерогенных документов микросервисов в документо-ориентированном хранилище. *Моделирование, оптимизация и информационные технологии.* 2022;10(4). Доступно по: https://moitvivt.ru/ru/journal/pdf?id=1247. DOI: 10.26102/2310-6018/2022.39.4.003 (на англ.) (дата обращения: 25.01.2023 г.).
33. Гусаренко А.С. Ситуационно-ориентированные базы данных: верификация

электронных подписей гетерогенных документов в RESTful веб-сервисе. *Моделирование, оптимизация и информационные технологии.* 2022;10(4). Доступно по: https://moitvivt.ru/ru/journal/pdf?id=1281. DOI: 10.26102/2310-6018/2022.39.4.019 (на англ.) (дата обращения: 25.01.2023 г.).

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Гусаренко Артем Сергеевич, кандидат технических наук, доцент кафедры Автоматизированных систем управления, Уфимского университета науки и технологий, Уфа, Российская Федерация.
e-mail: gusarenko.as@ugatu.su
ORCID: 0000-0003-4132-6106

Artem Sergeevich Gusarenko, Candidate of Technical Sciences, Associate Professor at the Department of Automated Control Systems, Ufa University of Science and Technology, Ufa, the Russian Federation.

Статья поступила в редакцию 12.02.2023; одобрена после рецензирования 21.03.2023; принята к публикации 10.04.2023.

The article was submitted 12.02.2023; approved after reviewing 21.03.2023; accepted for publication 10.04.2023.