УДК 004.65
DOI: 10.26102/2310-6018/2022.37.2.021

Situation-Oriented Databases: Processing Office Documents

V.V. Mironov, A.S. Gusarenko✉, N.I. Yusupova

Ufa State Aviation Technical University, Ufa, Russia
gusarenko.as@ugatu.su✉

Abstract. This article discusses the application of a situation-oriented approach to the problem of extracting semantic information from office documents. Office documents created by vector graphics editors and word processors are reviewed. The ability to extract semantic information is due to the fact that such documents are based on open XML formats that can be processed by external programs. Processing of documents based on a situational database where word documents are programmatically loaded as XML files extracted from zip-archives is considered. In the situation-oriented database, it is possible to present an office document as a virtual document that is mapped both on XML files and the ZIP archive with XML files. This applies not only to text documents, but also to graphic documents that have an internal XML representation. This enables processing of documents in Office Open XML and Open Document Format. The article discusses various aspects of identifying and finding the necessary information during document processing by means of special standard definitions as bookmarks, key phrases and text labels. Models and algorithms for extracting the required information are examined. Examples of the practical use of this approach in the field of distance learning of students at the university are given. In addition, an example of extracting metadata of scientific publications in the Open Journal Systems publishing system is regarded.

Keywords: situation-oriented database, built-in dynamic model, Office Open XML, Open Document Format.

Acknowledgements: This research is supported by RFBR (grant 19-07-00682). The results of the study, reflecting the structure of the developed software solution, were obtained as part of the state task No. FEUE-2020-0007.

For citation: 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 (In Eng.)

Ситуационно-ориентированные базы данных: обработка офисных документов

В.В. Миронов, А.С. Гусаренко✉, Н.И. Юсупова

Уфимский государственный авиационный технический университет, Уфа, Российская Федерация
gusarenko.as@ugatu.su✉

Резюме. В статье рассматривается подход построения документоориентированных веб-приложений на основе ситуационно-ориентированных баз данных. Приложения на базе ситуационно-ориентированных баз данных решают проблемы с извлечением и обработкой семантической информации из офисных документов. В уже имеющихся исследованиях рассматривались вопросы заполнения офисных документов, в данном же исследовании рассматриваются методы извлечения информации из графических документов и текстовых документов, созданных в обычных офисных пакетах. Создание и задействование таких методов достигается за счет характера внутреннего представления офисных документов в XML и возможности обработки такого содержимого программным способом. Рассматривается обработка XML-файлов в ситуационно-ориентированных базах данных, где Word-документы

программно загружаются как XML-файлы, извлекаемые из ZIP-архивов. В дальнейшем после загрузки документы могут быть представлены как виртуальные документы или множество таких документов, объединенных в виртуальный массив данных и отображаемых на реальные данные XML или ZIP-архивы с XML файлами внутри. Разработанные и применяемые методы работают в отношении как графических, так и текстовых документов. В статье также рассматриваются методы отыскания и идентификации нужных фрагментов данных внутри документа во время его обработки, базирующейся на стандартах описания в закладках, ключевых фразах, и текстовых метках. Модели и алгоритмы для извлечения требующейся информации обсуждаются и демонстрируются на практических примерах, где рассматривается система дистанционного выполнения курсовых проектов студентами. В дополнение к примерам из учебного процесса рассматривается извлечение метаданных научных публикаций из международной издательской системы Open Journal Systems.

***Ключевые слова*:** ситуационно-ориентированная база данных, встроенная динамическая модель, Office Open XML, Open Document Format.

Благодарности: Исследование выполнено при финансовой поддержке РФФИ (грант № 19-07-00682). Результаты исследования, отображённые в структуре разработанного программного решения, были получены в рамках государственного задания № FEUE-2020-0007.

***Для цитирования*:** Миронов В.В., Гусаренко А.С., Юсупова Н.И. Ситуационно-ориентированные базы данных: обработка офисных документов. *Моделирование, оптимизация и информационные технологии.* 2022;10(2). Доступно по: https://moitvivt.ru/ru/journal/pdf?id=1187 DOI: 10.26102/2310-6018/2022.37.2.021 (На англ.)

Introduction

Programmatic extraction of data from office documents is an information technology task that occurs quite often, especially in relation to documents created in the environment of Microsoft Visio graphics editor and its analogues, as well as Microsoft Word (a word processor) and its analogues (see the Microsoft Office package, OpenOffice, LibreOffice, Russian MyOffice, Chinese WPS Office and others). We will call such files office documents. In this article, we are interested in such office documents that have a "dual purpose": firstly, as an original beautifully designed document of polygraphed quality and, secondly, as a source of information for further automated processing.

Thus, the task of processing office documents arises when there is a need to extract the data of interest. To solve this problem, two questions need to be answered: 1) how to implement programmatic access to the content of the office document; 2) how to programmatically find the necessary information in the office document.

It is possible to programmatically process office documents because they are based on open formats such as the Office Open XML standard and the Open Document Format standard [1–3]. In these formats, the electronic document is organized in the form of a ZIP archive, in which folders with XML files are packed. Thus, the program can open the archive, extract the required XML file from it and process it as needed. For this, in principle, you can use different approaches: from direct work with XML files to the use of special software tools. In this article, this problem is solved using the platform of situational databases — an information processor that provides processing of heterogeneous data under the control of a hierarchical situational model [4, 5].

Literature review

The analysis of publications on this matter indicates an interest in the general problems of software processing of office documents — both in the conceptual plan of information retrieval and text processing [6–8] as well as in terms of practical application in different subject

areas [9–11] and the construction of software tools for solving specific problems [12–14]. Many publications discuss the technical aspects of programmatic processing of office documents based on OLE Automation [15,16], VBA, Word Object Model [17,18] and other tools [19,20]. Several publications are related to the programmatic filling of office documents templates-blanks with data [21–23] — this is the problem of entering data, which in its essence is the reverse of the problem of data extraction. As for the situational-oriented approach, it should be noted that the task [24] in which the problem of programmatic entering of data into the blank word documents as well as the task [25] where a similar problem is solved in relation to vector graphics documents, should be considered. At the same time, it was not possible to find any publications devoted to the issue of extracting the required information from office documents by processing them programmatically as text files structured with XML markup. Popular vector graphics editors are Microsoft Office Visio (proprietary) as well as its freely distributed counterparts OpenOffice Draw, LibreOffice Draw, etc. [26, 27]. Numerous publications reflect the effective application of these editors to create diagrams in a variety of subject areas: educational process [28, 29], software development and documentation [30], business process modeling [31], automated systems design [32], development and analysis of workflow models [33–35], database integration [6,7], query processing in natural language [39], etc. These editors are not tied to specific graphic notations; therefore, they can be utilized to define schemes in advanced provided notation. Another feature is the employment of open graphic formats to create the resulting diagrams. This makes it possible in principle to automatically process graphic documents to extract semantic properties from the schemas they contain. At the same time, as the analysis of publications has shown, the issues of extracting circuit properties are practically not covered in the literature (neither in technical nor in scientific aspects). There are no general principles or practical alternatives for solving this problem. This determines the relevance of the study on the extraction of information from office documents based on document processing [8,9].

Processing office documents in a situation-oriented environment

Situation-oriented databases are an approach that develops the idea of *polyglot persistence* — the exchange of heterogeneous data in one application [10]. This approach is based on the use of a built-in hierarchical situational model for managing the data processing (model-driven approach), as well as the concept of virtual documents mapped on real heterogeneous data (Figure 1) [40].

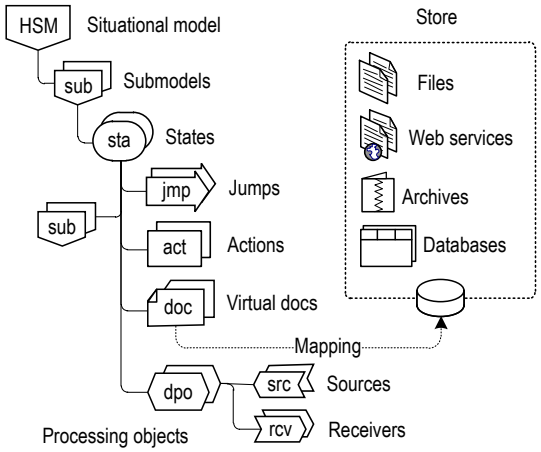

Figure 1 – A hierarchical situational model
Рисунок 1 – Иерархическая ситуационная модель

The Hierarchical Situation Model (`HSM`) defines data processing in a declarative form. It is a hierarchy of sub-models (`sub`) containing a set of states (`sta`) which can contain other sub-models (`sub`), transitions (`jmp`) and actions (`act`). Virtual documents (`doc`) together with processors (`dpo`) provide unified processing of heterogeneous external data. Virtual documents can be mapped to local files, remote web services, archives, relational databases. Based on the information contained in the declarations of virtual documents, external data is loaded into handlers — data processing objects (`dpo`) where they undergo the necessary transformations. The main types of handlers are DOM objects (Document Object Model) oriented towards processing XML data and associative arrays aimed at processing JSON data [6].

This article proposes an application of this approach since the extraction of data from office documents based on different formats and the subsequent transformation of the extracted data into a different format are in line with the general concept of uniform processing of heterogeneous data [8,9]. This, on the one hand, will extend the situational-oriented approach to office documents, on the other, it will simplify the processing of documents by applying a declarative hierarchical situational model.

Mapping to word-documents. Considering that in our task a word-document is a ZIP-archive, it is important, firstly, to be able to display virtual documents on archives. And the fact that XML files are in the archive folders determines the use of DOM objects as handlers. At the same time, you need to know the internal structure of both the ZIP-archive of the word-document and the XML-files in it. In Figure 2, this is illustrated for two word-document formats: for the DOCX format (Figure 2, a), which is used in the Microsoft Word editor environment, and for the ODT format (Figure 2, b), used in the OpenOffice Writer, LibreOffice Writer and other alternatives for MS Word.

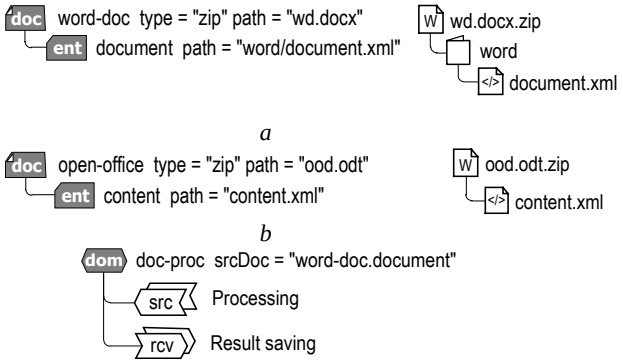

Figure 2 – Defining a virtual document with mapping on DOCX *(a)* and ODT *(b)* formats and processing in a DOM object *(c)*

Рисунок 2 – Задание виртуального документа в модели с отображением на DOCX *(a)* и ODT *(b)* обрабатываемых в DOM-объекте *(c)*

Search for the retrieved data in the xml tree of the document

Although for the internal presentation of office documents, open standardized formats are used, which are well documented, finding data in them is not a trivial task. This is due to the extremely complex internal structure of XML markup, in which the semantic information itself is dismembered and mixed with data that defines the external presentation of the document (fonts, indents, styles, structural elements, etc.). Therefore, it is necessary to study the internal structure of office documents to create models and methods focused specifically on the search and identification of the required data. Such research was carried out by direct examination of the internal XML-markup of the document with different ways of specifying the external presentation of data.

An XML component of an office document loaded into a DOM object is available for programmatic processing in the form of an XML tree containing nodes of XML elements and XML attributes. The task of finding the necessary nodes of the XML-tree, corresponding to the retrieved document data, arises. To do this, the retrieved data must satisfy the requirement of identifiability, namely, when entering data at the stage of creating a document, the user must comply with the requirements that make it possible to find this data in the future during program processing of the document. The following methods of ensuring data identifiability are proposed: key phrases; bookmarks; content controls; custom XML components.

Use of key phrases. To ensure identifiability by means of this method, certain keywords or phrases are entered into the data being sought and then the task of extracting is reduced to finding a paragraph (several paragraphs) containing this phrase. At the same time, it is necessary to consider the peculiarities of the internal XML-markup, determined by the format of the word-document being used. In DOCX format, each paragraph corresponds to a `w:p` element, in which one or more formatted `w:r` elements are nested, which, in turn, contains a `w:t` element. Here, the w prefix specifies that the XML element belongs to a specific namespace. The textual content of the entire paragraph is thus comprised of the textual content of all nested `w:t` elements. In ODT format, each paragraph has a `text:p` element that contains text nodes either directly or as text content of `text:span` elements with their own formatting. The text prefix specifies that the XML element belongs to the namespace. Thus, to extract the text content of a paragraph, you need to gather the text from all the internal text nodes of the corresponding `w:p` or `text:p` XML element. The text content of the paragraph, containing the key text, can be extracted from the XML tree using the appropriate XPath expression.

Using bookmarks. When employing this method, the user selects the desired section of text in the document by means of the so-called bookmark. Bookmarks can be pre-placed in the template-blank document and then the user enters data in the places tagged with the appropriate bookmarks. Since each bookmark has a unique name, the program can find and extract the selected text while processing the document. In the internal XML markup, each bookmark has two XML elements, a start element and an end element that are nested within a paragraph element. In DOCX format, these are XML elements `w:bookmarkStart` and `w:bookmarkEnd`, with each bookmark identified by an automatically assigned XML attribute `w:id` and a user-assigned `w:name` attribute. The ODT format utilizes the `text:bookmark-start` and `text:bookmark-end` elements, which are identified by the `text:name` attribute. The corresponding XPath expression can be applied to retrieve the text of a paragraph containing a bookmark with a specific name.

Using content controls. Content controls are container objects of data in a document that provide specific structuring and formatting. When used in document templates such elements as formatted and unformatted text, lists, and others are convenient both when entering data by users and when extracting this data by programs. Figure 3 shows the hierarchical models of the internal representation of content controls in DOCX and ODT formats. Models are built because of direct examination of XML markup and contain only those elements and attributes that are essential for the task at hand.

Applying custom XML components. Custom XML components are user XML data embedded in a document. Elements of a custom XML component can be bound to document content controls and then data entry or changes in the controls will be synchronously reflected in the custom XML component and vice versa. Figure 4 shows the hierarchical models of the internal representation of custom XML-parts in DOCX and ODT formats obtained by direct examination of XML markup.

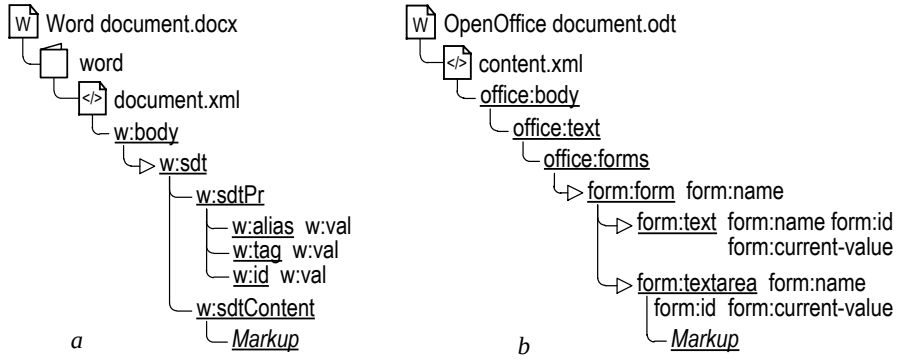

Figure 3 – Internal representation of content controls in DOCX *(a)* and ODT *(b)* formats
Рисунок 3 – Внутреннее представление управляющих элементов в DOCX *(a)* и ODT *(b)* форматах

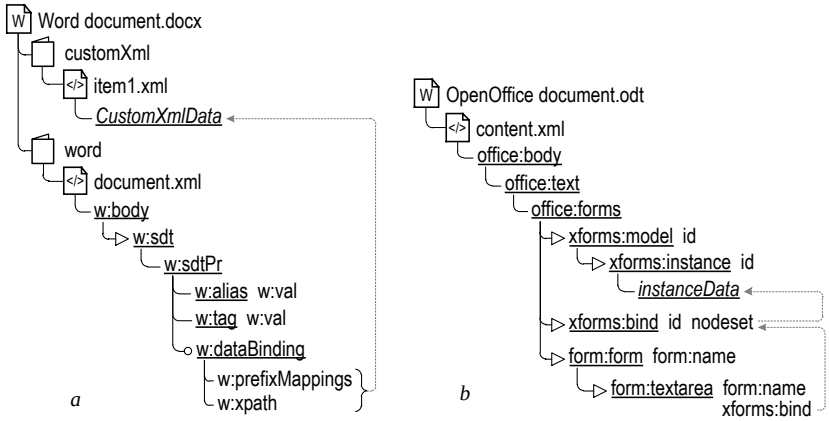

Figure 4 – Internal representation of custom XML parts in DOCX *(a)* and ODT *(b)* formats
Рисунок 4 – Внутреннее представление пользовательских XML-спецификаций в DOCX *(a)* и ODT *(b)* форматах

The choice of the identifiability method. Thus, there are several ways to ensure that retrievable text information is identifiable in office documents. The ability to use this or that method depends on the functional limitations of the editor used. The methods are characterized by varying degree of complexity from three interrelated points of view: firstly, from the point of view of the blank template developer; secondly, from the point of view of the user in terms of entering data, the template is a blank; thirdly, from a programmer's point of view in terms of programmatically extracting and cleaning data. Thus, the method of key phrases does not require preliminary efforts in terms of preparing a template for the filled-out document.

However, its implementation will require the user to follow strict rules in order to complete the document, which raises the possibility of errors. The rest of the methods demand some effort on the preliminary marking of the filled fields in the document template — the placement of bookmarks, controls. The application of controls can potentially be convenient for the user when filling out data due to a user-friendly interface and error control in input data. Employment of custom XML components is the most time-consuming approach in terms of template development: XML file with filled data elements has to be created and then embedded in the template as an XML component; content controls need to be placed in the displayed part of the document, and the binding of controls is to be configured to the corresponding elements of a custom XML component.

Retrieving data using this option is the easiest because the required data in the form of a custom XML component is already separated from the rest of the document.

Case study: uploading articles to a publishing system

The practical significance of the findings lies in the fact that they can serve as the basis for information technology for software processing of word documents in various fields. As a practical example of extracting text data from word documents based on a situationally oriented approach, we will consider this problem in relation to the process of preparing articles for publication in a scientific journal in the Open Journal System publishing platform (OJS, Figure 5). During the publishing process, the articles themselves along with their metadata are loaded into the system both upon initial submission and during revision and correction. In the traditional approach, these processes are duplicated: the next version of the article is uploaded to the site, containing metadata in its text, and, in addition, metadata is entered separately (by the authors and / or editorial staff). In this case, the metadata is entered manually: the next metadata element is copied from the article and keyed into the corresponding field the publishing system web interface.

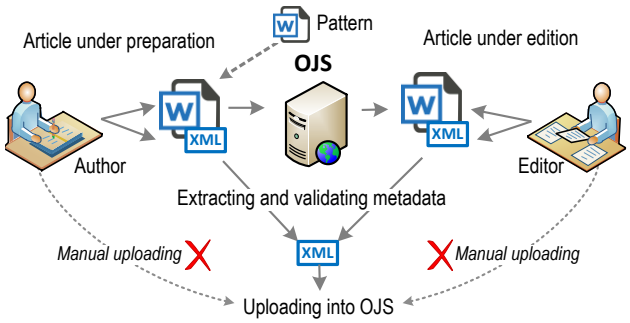

Figure 5 – Uploading papers to Open Journal System
Рисунок 5 – Загрузка статей в систему Open Journal System

Due to the large number of elements, this procedure is rather laborious. The goal is to reduce the complexity by automatically extracting metadata from the text of the article and loading it into the system. At the same time, it is also supposed to automate the verification of the extracted metadata to identify certain errors.

Hierarchical situational model. The metadata extraction procedure is implemented as a web application, the server side of which is launched by HTTP requests from clients. Figure 7 shows the `sub:Metadata-proc` HSM model, which provides the extraction and processing of metadata from a word document.

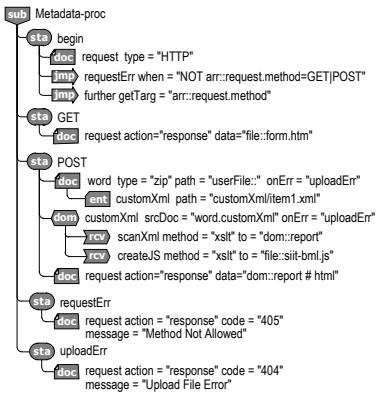

Figure 6 – Hierarchical situational model for extracting and processing metadata
Рисунок 6 – Иерархическая ситуационная модель для извлечения и обработки метаданных

Based on the proposed approach, a web application has been developed to automate the on-screen form filling of a publishing system with metadata from a scientific article. This web application is designed for automated extraction of metadata (titles, abstracts, keywords, bibliography, information about authors, etc.) from a scientific article and entering them into the fields of screen forms for uploading to the publishing system. A web application consists of a client and a server.

The client part includes the following components:

– A template for an article in the DOCX format for a word processor MS Word, containing a set of content controls for entering metadata.

– Article metadata base in XML format, embedded in the template-stub in the form of a custom XML-part, the elements of which are bound to content controls of the template-stub.

The server part includes the following components:

– A situational model in the HSM language, which helps to upload to the web server a prepared article based on a blank template, extracting metadata from it, starting the formation of the result.

– Script in PHP, providing the functionality of the situational model without using an HSM interpreter.

– Web form in HTML language for uploading an article to a web server.

– Style sheet in XSLT, which provides the formation of the resulting function in JavaScript for filling in the fields of the publishing system display forms.

The web application functions as follows. The author prepares an article by filling out a template, while adding metadata through content controls. Next, a GET request is sent through the web browser to the server side, in response to which a web form is displayed in the browser to download the article. The back end (HSM / PHP script) loads the article, extracts the custom XML metadata part from it, and XSLT transforms it according to the stylesheet. The result is a metadata validation report that is returned to the client's web browser and a generated JavaScript function that is saved as a file on the web server. The metadata is loaded into the publishing system by a bookmarklet hosted in the client's web browser and referencing a JavaScript function stored on the web server. When publishing forms are open in a browser, the client user runs a bookmarklet that downloads from the server and applies a JavaScript function to the forms. As a result, the form input fields are filled with the appropriate metadata.

Using the web application frees the author and/or editor of the journal from manually entering metadata into the publishing system. Because articles contain a significant number of metadata, automation significantly reduces the complexity of this process.

This web application is used to publish the scientific journal "Systems Engineering and Information Technologies" (http://siit.ugatu.su) on the Open Journal System (OJS) platform. It can be easily configured for application in other magazines and publishing platforms.

Extracting information from graphics documents

Let us consider the task of extracting semantic information from a chart contained in an electronic document created in a graphical editor in vector graphics format. Semantic information here means the structural and parametric properties of the diagram, cleared of insignificant details of its visual representation such as the size of figures, line thickness, fill color, etc. The solution of the problem should be a graph whose vertices correspond to the elements of the diagram and arcs — connections of elements with each other. Vertices and arcs must be loaded with the parameters specified in the scheme. In the technical aspect, the task is to programmatically extract the structural and parametric properties of the graphic scheme of interest and enter them into the database for subsequent use. In this case, two levels of presenting information in a graphical document should be distinguished: external,

corresponding to the visual display of the scheme, and internal, corresponding to the internal coding of the elements of the chart according to the used graphical format.

The idea of solving the problem is to explore the specifics of assigning symbols at the internal level in various formats of vector graphics and, on this basis, learn how to identify the symbols used in a particular chart. The proposed approach is generally illustrated in Figure 7.

Conceptual model. By the conceptual model of the diagram, we mean the model of the upper level of abstraction, which specifies a set of elements, their properties as well as relationships with each other. Each element of the diagram is an instance of a conditional graphic image.

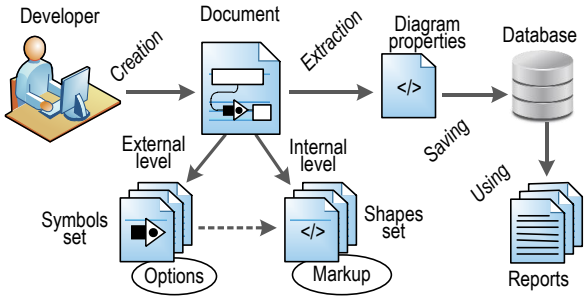

Figure 7 – Extracting semantic information from a graphic document
Рисунок 7 – Извлечение семантической информации из графического документа

The conceptual model defines which elements are present in the diagram; what the properties (options, parameters) of these instances are; how instances are nested within each other; how the instances are interconnected. In Figure 8, on the right, the conceptual model is illustrated using a tree where the Schema root contains many children (`Element`) and many child connections (`Connect`) which, in turn, contain many properties (`Property`). The nesting relationship of one element within another is specified by an optional reference to the parent (`Parent`). The relationship of connecting one element to another is specified by a pair of references to these elements (`From` and `To`) contained in the connection. Thus, the conceptual model reflects the structure of what is to be obtained from the extraction.

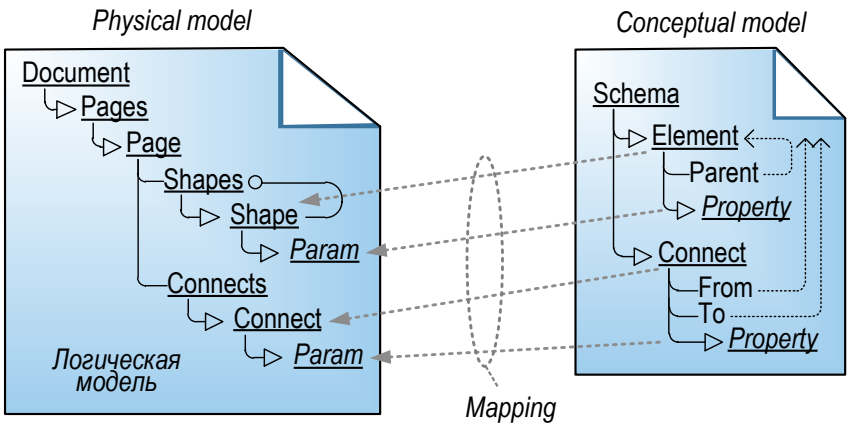

Figure 8 – Conceptual-logical mapping
Рисунок 8 – Концептуально-логическое отображение

Identification of chart elements. When processing a graphic document, it is necessary to enumerate its constituent figures to identify the elements of the circuit. This process depends,

firstly, on the graphic format used, and secondly, on the specific representation of the shapes of the conventional graphic symbols. Thus, the problem of constructing identification models for circuit elements for specific graphic formats arises. Such models are obtained because of direct research of the internal XML-markup of graphic documents for the open formats of Visio, LibreOffice Draw and OpenOffice Draw editors.

Practical use of the results. The approach given above was tested and applied in the educational process at Ufa State Aviation Technical University to extract the properties of graphic models during course design. Depending on the problem being solved, the following information is extracted from this graphic document:

– Structure of the relational model – used to automatically check its syntactic correctness as well as compliance with the task and the results of the previous stages of development.

– Program code text – is used to automatically check the correspondence of the SQL code compiled by the student to the relational model diagram as well as to automatically generate draft design documentation (for example, a "program text" document).

Metadata from the title block of the document – for use in reports and project documentation. As a result, a noticeable decrease in routine actions is achieved to control the correctness of the drawn-up scheme and the preparation of reporting documentation both for teachers and students.

Conclusion

To conclude, this article deals with the application of a situational-oriented approach for programmatic processing of office documents. Namely, documents prepared by the user in the Microsoft Visio graphics editor and its analogues as well as in the environment of Microsoft Word and its analogues. Documents are used hereinafter as information sources. The openness of the Office Open XML and Open Document Format made it possible to apply the concept of virtual documents mapped to ZIP archives for programmatic access to XML components of word documents in a situational environment. The importance of developing preliminary agreements regarding the placement of information in a document for subsequent search and retrieval, for example, using pre-prepared templates, has been substantiated. For the DOCX and ODT formats, the use of key phrases, bookmarks, content controls, custom XML components to organize the extraction of the entered data is considered. For each option, tree-like models of access to the extracted data have been built. It is noted that the application of one or another option depends on the functionality and limitations of a graphics editor or word processor and is characterized by varying degrees of a blank template development, entering data by the user and programming for data extraction. A practical example of processing the metadata from a scientific article prepared in Microsoft Word for publication in a scientific journal is assessed. The applied solution is based on entering metadata into the article using content controls placed in a stub template and bound to elements of a custom XML component. The developed hierarchical situational model of HSM provides extraction of an XML component loading it into a DOM object and XSLT transformations to obtain the resulting data: an error report and JavaScript code for the subsequent use of the extracted metadata. The results have found practical application in the publication of a scientific journal and in the system of distance learning at university.

REFERENCES

1. Hou X., Li N., Yang H., Liang Q. Comparison of Wordprocessing Document Format in OOXML and ODF. In: 2010 Sixth International Conference on Semantics, Knowledge and Grids. 2010:297–300. DOI:10.1109/SKG.2010.44

2. Schubert S. The Next Millennium Document Format. In DocEng'19: Proceedings of the ACM Symposium on Document Engineering 2019. New York, NY, USA: Association for Computing Machinery. 2019:1–4. DOI:10.1145/3342558.3345419
3. Roig J., Ribera M. Implementation of the OOXML standard since its approval until today. In DSAI'2020: 9th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion. New York, NY, USA: Association for Computing Machinery. 2020:129–134. DOI:10.1145/3439231.3440607
4. Mironov V.V., Gusarenko A.S., Yusupova N.I. Structuring virtual multi-documents in situationally-oriented databases by means of entry-elements. SPIIRAS Proceedings. 2017;4(53):225–242. DOI:10.15622/sp.53.11 (In Russ.)
5. Mironov V.V., Gusarenko A.S., Yusupova N.I. Situation-oriented databases: polyglot persistence based on REST microservices. Applied Informatics. 2019;5(83):87–97. DOI:10.24411/1993-8314-2019-10038 (In Russ.)
6. Mironov V.V., Gusarenko A.S., Yusupova N.I., Smetanin Y.G. JSON documents processing using situation-oriented databases. Acta Polytechnica Hungarica. 2020;17(8):29–40. DOI:10.12700/APH.17.8.2020.8.3
7. Mironov V.V., Gusarenko A.S., Tuguzbaev G.A. Graphic Documents Parametric Personalization for Information Support of Educational Design Using Situation-Oriented Databases. In ITIDS'2020: 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support. Atlantis Press. 2020:260–267. DOI:10.2991/assehr.k.201029.050
8. Mironov V.V., Gusarenko A.S., Tuguzbaev G.A. Extracting semantic information from graphic schemes. Informatics and Automation. 2021;20(4):940–970. DOI:10.15622/IA.20.4.7 (In Russ.)
9. 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*. 1204 CCIS. 2021:167–178. DOI:10.1007/978-3-030-78273-3_17
10. 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. *SIIT*. 2021;3(1(5)):39–49.
11. Kulkarni A., Shivananda A. *Extracting the Data – Natural Language Processing Recipes*. Springer; 2019.
12. Bolotova LS, Danchul AN, Novikov AP, Surkhaev MA, Nikishina AA. Initial identification in technology of informational search (part 1). *Prikladnaya Informatika = Journal of Applied Informatics.* 2015;4(58):128–142.
13. Bolotova L.S., Danchul A.N., Novikov A.P., Surkhaev M.A., Nikishina A.A. Initial identification in technology of informational search (part 2). *Prikladnaya Informatika = Journal of Applied Informatics*. 2015;6(60):128–143.
14. Joun J., Chung H., Park J., Lee S. Relevance analysis using revision identifier in MS word. *Journal of Forensic Sciences*. 2021;66(1):323–335.
15. Jarzabek S., Dan D. Documentation Reuse: Managing Similar Documents. *In: 2017 IEEE International Conference on Information Reuse and Integration (IRI).* 2017:372–375. DOI:10.1109/IRI.2017.52
16. Bešić D. Microservice for text extraction from word and pdf documents. *In: Proceeding of the Faculty of technical Sciences, Novi Sad.* 2021;36(07):1252–1256. DOI:10.24867/13BE26Besic
17. Duretec K., Rauber A., Becker C. A Text Extraction Software Benchmark Based on a Synthesized Dataset. *In: 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL).* 2017:1–10. DOI:10.1109/JCDL.2017.7991565

18. Karcioğlu A.A., Yaşa A.C. Automatic Summary Extraction in Texts Using Genetic Algorithms. *In: 2020 28th Signal Processing and Communications Applications Conference (SIU).* 2020:1–4. DOI:10.1109/SIU49456.2020.9302205
19. Harmata S., Hofer-Schmitz K., Nguyen P.H., Quix C., Bakiu B. Layout-Aware Semi-automatic Information Extraction for Pharmaceutical Documents. *In: Da Silveira M, Pruski C, Schneider R, editors. Data Integration in the Life Sciences.* Cham: Springer International Publishing. 2017:71–85. (Lecture Notes in Computer Science). DOI:10.1007/978-3-319-69751-2_8
20. Zhang J., Xie Y., Shen J., Wang L., Lin H. Text Information Hiding Method Using the Custom Components. *In: Sun X, Pan Z, Bertino E, editors. Cloud Computing and Security. Cham: Springer International Publishing*. 2018:473–84. (Lecture Notes in Computer Science). DOI:10.1007/978-3-030-00015-8_41
21. Lubenets Y.V., Miroshnikov A.I. Software Supports for Remote Examination on Mathematical Disciplines in Higher Education. *In TBLE: 2021 1st International Conference on Technology Enhanced Learning in Higher Education*. 2021:274–277. DOI:10.1109/TELE52840.2021.9482472
22. Abramova I.A., Syrkin V.V., Stepanov A.P. Extensions of the standard functionality and interface of MS Office applications based on the development of custom add-ins. *Nauka i Voennaya Bezopasnost'*. 2020;2(21):192–199.
23. Miroshnikova E.P., Levonevskiy D.K., Motienko A.I. Modules for import, export and data analytics in the electronic journal management system of the 'Spiiras Proceedings' journal for automated interaction with global indices and aggregators. *Problemy iskusstvennogo intellekta = Problems of Artificial Intelligence.* 2019;3(14):58–75.
24. Reznichenko O.S., Sivakov S.I., Reznichenko T.A. Method of automated generation of information about university's scientific publications for reporting in the research management system of the russian ministry of science and higher education. *Universitetskoe Upravlenie: Praktika i Analiz = University Management: Practice and Analysis.* 2020;24(2):44–58. DOI: 10.15826/umpa.2020.02.013
25. Pinto J., Rathod D., and Quadros A. Text summarizer for URL and .DOCX files. *International Journal of Advanced Research in Computer Science*. 2020;11(4):18–21. DOI: 10.26483/ijarcs.v11i4.6639
26. Baynova M.S., Sokolov A.M. Tools for automated collection and analysis of sociological information on the territorial identity of city residents. *Prikladnaya Informatika = Journal of Applied Informatics*. 2021;2(92):92–102.
27. Novikov A., Keyno P. Heterogenius data collecting in scientific communities using portfolio management system in ConfID service. *Prikladnaya Informatika = Journal of Applied Informatics.* 2020; 2(86):28–36.
28. Izmailov V.V., Novoselova M.V. Automated system for generating task options based on MS Word document. *Software Journal: Theory and Applications.* 2017;1:1–5. DOI:10.15827/2311-6749.17.1.1
29. Yu Z., Xiong Z. Comparative analyses for the performance of Rational Rose and Visio in software engineering teaching. *In: J. Physics: Conf. Series, IOP Publishing*. 2018;1087(6):062–041. DOI:10.1088/1742-6596/1087/6/062041
30. Parker D.J. *Mastering Data Visualization with Microsoft Visio Professional 2016*. Packt Publishing Ltd; 2016.
31. He L., Lian J. Instructional design of practice course of logistics system planning and design based on Visio. *In ITME'2018: Proc. 9th Int. Conf. on Information Technology in Medicine and Education*. 2018:526–530. DOI:10.1109/ITME.2018.00122
32. Ruiz Ledesma E.F. et al. Educational tool for generation and analysis of multidimensional modeling on data warehouse. *Int. J. Advanced Computer Science and Applications*.

2020;11(9):261–267. DOI:10.14569/IJACSA.2020.0110930
33. Shafiee S. et al. Evaluating the benefits of a computer-aided software engineering tool to develop and document product configuration systems. *Computers in Industry*. 2021;128. DOI:10.1016/j.compind.2021.103432
34. Medoh C., Telukdarie A. Business process modelling tool selection: a review. *In IEEM'2017: Proc. IEEE Int. Conf. on Industrial Engineering and Engineering Management*. IEEE;2017;524–528. DOI:10.1109/IEEM.2017.8289946
35. Afanasyev A., Voit N., Gaynullin R. The analysis of diagrammatic of workflows in design of the automated systems. *In: Uncertainty Modelling in Knowledge Engineering and Decision Making*. 2016:509–514. DOI:10.1142/9789813146976_0082
36. Voit N., Bochkov S., Kirillov S. Temporal automaton RVTI-grammar for the diagrammatic design workflow models analysis. In AICT'2020: IEEE 14th Int. Conf. on Application of Information and Communication Technologies, Tashkent, Uzbekistan. 2020:1–6. DOI:10.1109/AICT50176.2020.9368810
37. Afanasyev A., Voit N., Ukhanova M., Ionova I. Development of the approach to check the correctness of workflows. *In: Data Science and Knowledge Engineering for Sensing Decision Support.* P. 1392–1399. DOI:10.1142/9789813273238_0173
38. Shah R., Kesan J. Interoperability challenges for open standards: ODF and OOXML as examples. *In dg.o'09: Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government.* Puebla: Digital Government Society of North America. 2009:56–62.
39. Doncevic J., Fertalj K. Database integration systems. *In MIPRO'2020: Proc. 43rd Int. Convention on Information, Communication and Electronic Technology*. 2020:1617–1622. DOI:10.23919/MIPRO48935.2020.9245245
40. 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
41. Kosmerl I., Rabuzin K., Sestak M. Multi-model databases – introducing polyglot persistence in the big data world. *In MIPRO'2020: Proc. 43rd Int. Convention on Information, Communication and Electronic Technology*. 2020:1724–1729. DOI:10.23919/MIPRO48935.2020.9245178
42. Montgomery C., Isah H., Zulkernine F. Towards a natural language query processing system. *In IBDAP'2020: Proc. 1st Int. Conf. on Big Data Analytics and Practices*. 2020. DOI:10.1109/IBDAP50342.2020.9245462

СПИСОК ИСТОЧНИКОВ

1. Hou X., Li N., Yang H., Liang Q. Comparison of Wordprocessing Document Format in OOXML and ODF. In: *2010 Sixth International Conference on Semantics, Knowledge and Grids.* 2010:297–300. DOI:10.1109/SKG.2010.44
2. Schubert S. The Next Millennium Document Format. In DocEng'19: Proceedings of the ACM Symposium on Document Engineering 2019. New York, NY, USA: *Association for Computing Machinery.* 2019:1–4. DOI:10.1145/3342558.3345419
3. Roig J., Ribera M. Implementation of the OOXML standard since its approval until today. *In DSAI'2020: 9th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*. New York, NY, USA: Association for Computing Machinery. 2020:129–134. DOI:10.1145/3439231.3440607
4. Миронов В.В., Гусаренко А.С., Юсупова Н.И. Структурирование виртуальных мультидокументов в ситуационно-ориентированных базах данных с помощью

entry-элементов. *Труды СПИИРАН*. 2017;(53):225–43. DOI:10.15622/sp.53.11
5. Миронов В.В., Гусаренко А.С., Юсупова Н.И. Ситуационно-ориентированные базы данных: polyglot persistence на основе REST-микросервисов. *Прикладная информатика*. 2019;14(5(83)):87–97. DOI:10.24411/1993-8314-2019-10038
6. Mironov V.V., Gusarenko A.S., Yusupova N.I., Smetanin Y.G. JSON documents processing using situation-oriented databases. *Acta Polytechnica Hungarica*. 2020;17(8):29–40. DOI:10.12700/APH.17.8.2020.8.3
7. Mironov V.V., Gusarenko A.S., Tuguzbaev G.A. Graphic Documents Parametric Personalization for Information Support of Educational Design Using Situation-Oriented Databases. *In ITIDS'2020: 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support*. Atlantis Press. 2020:260–267. DOI:10.2991/assehr.k.201029.050
8. Миронов В.В., Гусаренко А.С., Тугузбаев Г.А. Извлечение семантической информации из графических схем. *Информатика и автоматизация*. 2021;20(4):940–70. DOI:10.15622/IA.20.4.7
9. 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*. 1204 CCIS. 2021:167–178. DOI:10.1007/978-3-030-78273-3_17
10. 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. *SIIT*. 2021;3(1(5)):39–49.
11. Kulkarni A., Shivananda A. *Extracting the Data – Natural Language Processing Recipes*. Springer; 2019.
12. Bolotova LS, Danchul AN, Novikov AP, Surkhaev MA, Nikishina AA. Initial identification in technology of informational search (part 1). *Prikladnaya Informatika = Journal of Applied Informatics.* 2015;4(58):128–142.
13. Bolotova L.S., Danchul A.N., Novikov A.P., Surkhaev M.A., Nikishina A.A. Initial identification in technology of informational search (part 2). *Prikladnaya Informatika = Journal of Applied Informatics*. 2015;6(60):128–143.
14. Joun J., Chung H., Park J., Lee S. Relevance analysis using revision identifier in MS word. *Journal of Forensic Sciences*. 2021;66(1):323–335.
15. Jarzabek S., Dan D. Documentation Reuse: Managing Similar Documents. *In: 2017 IEEE International Conference on Information Reuse and Integration (IRI).* 2017:372–375. DOI:10.1109/IRI.2017.52
16. Bešić D. Microservice for text extraction from word and pdf documents. *In: Proceeding of the Faculty of technical Sciences, Novi Sad.* 2021;36(07):1252–1256. DOI:10.24867/13BE26Besic
17. Duretec K., Rauber A., Becker C. A Text Extraction Software Benchmark Based on a Synthesized Dataset. *In: 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL).* 2017:1–10. DOI:10.1109/JCDL.2017.7991565
18. Karcioğlu A.A., Yaşa A.C. Automatic Summary Extraction in Texts Using Genetic Algorithms. *In: 2020 28th Signal Processing and Communications Applications Conference (SIU).* 2020:1–4. DOI:10.1109/SIU49456.2020.9302205
19. Harmata S., Hofer-Schmitz K., Nguyen P.H., Quix C., Bakiu B. Layout-Aware Semi-automatic Information Extraction for Pharmaceutical Documents. *In: Da Silveira M, Pruski C, Schneider R, editors. Data Integration in the Life Sciences.* Cham: Springer International Publishing. 2017:71–85. (Lecture Notes in Computer Science). DOI:10.1007/978-3-319-69751-2_8
20. Zhang J., Xie Y., Shen J., Wang L., Lin H. Text Information Hiding Method Using the

Custom Components. *In: Sun X, Pan Z, Bertino E, editors. Cloud Computing and Security. Cham: Springer International Publishing*. 2018:473–84. (Lecture Notes in Computer Science). DOI:10.1007/978-3-030-00015-8_41

21. Lubenets Y.V., Miroshnikov A.I. Software Supports for Remote Examination on Mathematical Disciplines in Higher Education. *In TBLE: 2021 1st International Conference on Technology Enhanced Learning in Higher Education*. 2021:274–277. DOI:10.1109/TELE52840.2021.9482472
22. Abramova I.A., Syrkin V.V., Stepanov A.P. Extensions of the standard functionality and interface of MS Office applications based on the development of custom add-ins. *Nauka i Voennaya Bezopasnost'*. 2020;2(21):192–199.
23. Miroshnikova E.P., Levonevskiy D.K., Motienko A.I. Modules for import, export and data analytics in the electronic journal management system of the 'Spiiras Proceedings' journal for automated interaction with global indices and aggregators. *Problemy iskusstvennogo intellekta = Problems of Artificial Intelligence*. 2019;3(14):58–75.
24. Reznichenko O.S., Sivakov S.I., Reznichenko T.A. Method of automated generation of information about university's scientific publications for reporting in the research management system of the russian ministry of science and higher education. *Universitetskoe Upravlenie: Praktika i Analiz = University Management: Practice and Analysis.* 2020;24(2):44–58. DOI: 10.15826/umpa.2020.02.013
25. Pinto J., Rathod D., and Quadros A. Text summarizer for URL and .DOCX files. *International Journal of Advanced Research in Computer Science*. 2020;11(4):18–21. DOI: 10.26483/ijarcs.v11i4.6639
26. Baynova M.S., Sokolov A.M. Tools for automated collection and analysis of sociological information on the territorial identity of city residents. *Prikladnaya Informatika = Journal of Applied Informatics*. 2021;2(92):92–102.
27. Novikov A., Keyno P. Heterogenius data collecting in scientific communities using portfolio management system in ConfID service. *Prikladnaya Informatika = Journal of Applied Informatics.* 2020; 2(86):28–36.
28. Izmailov V.V., Novoselova M.V. Automated system for generating task options based on MS Word document. *Software Journal: Theory and Applications.* 2017;1:1–5. DOI:10.15827/2311-6749.17.1.1
29. Yu Z., Xiong Z. Comparative analyses for the performance of Rational Rose and Visio in software engineering teaching. *In: J. Physics: Conf. Series, IOP Publishing.* 2018;1087(6):062–041. DOI:10.1088/1742-6596/1087/6/062041
30. Parker D.J. *Mastering Data Visualization with Microsoft Visio Professional 2016*. Packt Publishing Ltd; 2016.
31. He L., Lian J. Instructional design of practice course of logistics system planning and design based on Visio. *In ITME'2018: Proc. 9th Int. Conf. on Information Technology in Medicine and Education*. 2018:526–530. DOI:10.1109/ITME.2018.00122
32. Ruiz Ledesma E.F. et al. Educational tool for generation and analysis of multidimensional modeling on data warehouse. *Int. J. Advanced Computer Science and Applications*. 2020;11(9):261–267. DOI:10.14569/IJACSA.2020.0110930
33. Shafiee S. et al. Evaluating the benefits of a computer-aided software engineering tool to develop and document product configuration systems. *Computers in Industry*. 2021;128. DOI:10.1016/j.compind.2021.103432
34. Medoh C., Telukdarie A. Business process modelling tool selection: a review. *In IEEM'2017: Proc. IEEE Int. Conf. on Industrial Engineering and Engineering Management*. IEEE;2017;524–528. DOI:10.1109/IEEM.2017.8289946
35. Afanasyev A., Voit N., Gaynullin R. The analysis of diagrammatic of workflows in design of the automated systems. *In: Uncertainty Modelling in Knowledge Engineering*

and Decision Making. 2016:509–514. DOI:10.1142/9789813146976_0082
36. Voit N., Bochkov S., Kirillov S. Temporal automaton RVTI-grammar for the diagrammatic design workflow models analysis. In AICT'2020: IEEE 14th Int. Conf. on Application of Information and Communication Technologies, Tashkent, Uzbekistan. 2020:1–6. DOI:10.1109/AICT50176.2020.9368810
37. Afanasyev A., Voit N., Ukhanova M., Ionova I. Development of the approach to check the correctness of workflows. *In: Data Science and Knowledge Engineering for Sensing Decision Support.* P. 1392–1399. DOI:10.1142/9789813273238_0173
38. Shah R., Kesan J. Interoperability challenges for open standards: ODF and OOXML as examples. *In dg.o'09: Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government.* Puebla: Digital Government Society of North America. 2009:56–62.
39. Doncevic J., Fertalj K. Database integration systems. *In MIPRO'2020: Proc. 43rd Int. Convention on Information, Communication and Electronic Technology*. 2020:1617–1622. DOI:10.23919/MIPRO48935.2020.9245245
40. 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
41. Kosmerl I., Rabuzin K., Sestak M. Multi-model databases – introducing polyglot persistence in the big data world. *In MIPRO'2020: Proc. 43rd Int. Convention on Information, Communication and Electronic Technology*. 2020:1724–1729. DOI:10.23919/MIPRO48935.2020.9245178
42. Montgomery C., Isah H., Zulkernine F. Towards a natural language query processing system. *In IBDAP'2020: Proc. 1st Int. Conf. on Big Data Analytics and Practices*. 2020. DOI:10.1109/IBDAP50342.2020.9245462

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Миронов Валерий Викторович, д.т.н., профессор кафедры Автоматизированных систем управления, Уфимского государственного авиационного технического университета, Уфа, Российская Федерация.
e-mail: mironov@list.ru
ORCID: 0000-0002-0550-4676

Valeriy Viktorovich Mironov, Doctor of Technical Sciences, Professor of the Department of Automated Control Systems, Ufa State Aviation Technical University, Ufa, Russian Federation.

Гусаренко Артем Сергеевич, к.т.н., доцент кафедры Автоматизированных систем управления, Уфимского государственного авиационного технического университета, Уфа, Российская Федерация.
e-mail: gusarenko.as@ugatu.su
ORCID: 0000-0003-4132-6106

Artem Sergeevich Gusarenko, Candidate of Technical Sciences, Associate Professor of the Department of Automated Control Systems, Ufa State Aviation Technical University, Ufa, Russian Federation.

Юсупова Нафиса Исламовна, д.т.н., профессор кафедры Вычислительной математики и кибернетики, Уфимского государственного авиационного технического университета, Уфа, Российская Федерация.
e-mail: yusupova.ni@ugatu.su
ORCID: 0000-0002-7114-7638

Nafisa Islamovna Yusupova, Doctor of Technical Sciences, Professor of the Department of Computational Mathematics and Cybernetics, Ufa State Aviation Technical University, Ufa, Russian Federation.

Статья поступила в редакцию 19.05.2022; одобрена после рецензирования 06.06.2022; принята к публикации 28.06.2022.

The article was submitted 19.05.2022; approved after reviewing 06.06.2022;accepted for publication 28.06.2022.