Web Tool for Automated Document Formatting Verification

Viacheslav Martsinkevich
ITMO University
Saint Petersburg, Russian Federation
slavamarcin@yandex.ru

Vladislav Tereshchenko
ITMO University
Saint Petersburg, Russian Federation
vlad-tershch@yandex.ru

Galina Larionova
ITMO University
Saint Petersburg, Russian Federation
larionovags@yandex.ru

Andrey Berezhkov
ITMO University
Saint Petersburg, Russian Federation
berezhkov@itmo.ru

Elizaveta Kobets
ITMO University
Saint Petersburg, Russian Federation
www.kobets@yandex.com

Nail Nasyrov
Radio Engineering College
Saint Petersburg, Russian Federation
pasdel@mail.ru

Nataliya Gorlushkina
ITMO University
Saint Petersburg, Russian Federation
nagor@itmo.ru

***Abstract*—The article discusses the algorithms for document compilation check in compliance with standards and regulatory documents. The paper presents various approaches and methods of extraction of structural elements and their properties from PDF, ODT, DOCX documents; reveals machine-learning opportunities in terms of class extraction and further structural elements classification. The machine-learning methods are also used to provide recommendations on possible errors. The article introduces automated document formatting verification service architecture.**

I. Introduction

It worth mentioning that educational institutions mostly employ non-automated or partially automated methods, dealing with documents compilation and formatting check. If any report is needed, then the data systems are usually employed to create a document template. Such systems help to avoid some errors but are unable to check formatting of the report content.

In this study, the formatting verification means the check of documents formatting in terms of compliance with regulatory standards. Reports formatting verification ensures accuracy and consistency of information provided by learners, helps to avoid formatting and presentation errors that may occur during papers submission, ensures regulatory standards compliance.

The documents formatting verification automation has been regarded by many authors. Some of them have found a partial solution for this issue [1], [2]. The main advantage of documents formatting verification automation is a significant reduce in time, spent on documents check [3], [4].

At the same time, even a partial automation of documents formatting verification helps to ease the process, reduce errors and diminish human factor impact for the final document versions. We can use both deterministic approaches (such as regular expressions and static rules) and machine learning patterns to verify, if the structural elements formatting complies with the regulatory documents standards. The machine-learning patterns are able to create recommendations on proper formatting [5].

The documents formatting verification automation includes the following tasks:

1) To extract structural elements from the documents and their properties.
2) To ensure that each structural element complies with the formatting rules.
3) To make a report on discrepancies between structural elements and formatting rules.

Such sequence of actions may provoke some errors. For example, the structural element formatting fails to meet some rule.

The second issue appears, when we extract the same structural elements for different kinds of documents. As the properties storage methods are different, the extraction results also may differ from each other significantly.

The authors of this paper have developed a solution to resolve all the issues, mentioned above and to improve documents formatting verification approach. This is possible due to documents properties unification and use of pattern for structural elements classification.

The aim of this paper is to dwell on the development process of project, devoted to parsing methods application and extraction of structural elements and their properties for ODT, DOCX and PDF reporting documents. The project is aimed at unification and classification in order to predict the class of any element by means of open-source development methods.

The article reveals the parsing procedure for different document formats, shows the different ways of service scheme organization, dwells on the application of machine-learning methods for structural elements classification. The paper also discusses opportunities to improve service by means of the NLP.

II. Related works

If any report is needed, then the data systems are usually employed to create a document template. Such systems help to avoid some errors but are unable to check formatting of the report content. The examples of such systems are “Software suite for quicker formatting verification” (certificate RU 2019662634), “Student papers formatting verification» (RU 2015615893), “Automated system “Adequacy check” (RU 2015613397), StyleEase [6], DocVerifier (https://www.docverifier.com), macro package "Guidelines" (https://mtdmacro.ru/), MAPDoc [7], Killer-antiplagiat (https://killer-antiplagiat.ru/), “Document review module for software suite for automated documents formatting compliance check “The easiest way to perform an formatting verification” (RU2021680767).

The major problem of extraction of structural elements and their properties is the absence of a unified tool for parsing of various kinds of documents. However, the issues, associated with the parsing process of certain types of files have been frequently discussed. The authors of the following papers [8], [9], [10], have made significant contribution to the solution of this problem. They have thoroughly studied both the structural elements extraction process and opportunities for text thematic modulation. The authors of the mentioned works have concluded that PDF is one of the most complex formats in terms of structural elements data extraction.

The authors have already dwelled on the formats for which it is possible to extract structural elements and their properties [11].

ODT formats issues have been considered in the following articles [12], [13], [14]. The major challenges for ODT parsing are high nesting level, various ways of creating ODT files. That may result in various styles attributes.

DOCX parsing opportunities have been considered by the following authors [15], [16].

However, most cases are related to the lists, tables or paragraph content extraction. At the same time, the extraction of structural elements styles has remained almost intact.

The papers analysis has driven us to the conclusion that there are many unsettled problems in the field of document parsing. This is particularly relevant for PDF parsing. There are some partial solutions for DOCX for instance [17], but it is rather difficult to combine all of them and create a unified tool.

However, many authors, who support similar solutions, actively use machine-learning methods to obtain data on structural elements of a document, especially when they deal with formulae, figures, and headers [18], [19], [20]. Nevertheless, the unified parsing tool that is able to perform parsing for different document formats as well as to check the formatting by dynamic checking rules hasn’t been created yet.

III. Parsing methods and analysis

Parsing of elements in ODT documents means the extraction of parent styles and associated descendants. The object styles present a three-leveled hierarchy: default styles, text editors styles and automatic styles. The attributes are extracted from this retrieved data. The extracted attributes are thoroughly analyzed during documents formatting verification.

DOCX parsing is the simplest one, because it belongs to a standard style according to the style hierarchy. The detailed guidelines on DOCX parsing make it easier.

The parsing of structural elements and attributes in PDF documents includes several processes.

- Text symbols parsing, parsing of pdf objects, such as figures, fonts, and their attributes directly from PDF document text.
- Making up lines out of symbols and further paragraphs by a special algorithm.
- Creating tables from pdf objects.

Each process is associated with the creation of new properties in comparison with the original properties assigned to each element.

To implement a unified class the authors of this paper have elaborated data classes for Python, which describe the unified structure that is necessary for document elements. A class set comprises structures for each type of elements. These structures keep the elements content (a paragraph, a table, a figure, etc.) and their properties. As each document format has its own ways of keeping data on properties, the properties should be standardized. This will provide the opportunity for further analysis and classifier use. We should take into consideration the fact the elements that belong to the same class may have different formatting.

The study has identified more than 20 features that describe a unified structure. Apart from such evident parameters as font size, typeface size, indentation, etc. the more specific features that may influence element classification have also been mentioned (including those obtained by machine learning methods). To such properties belong, for example, first word trait, last symbol trait, text content length trait, the previous and the following element, etc.

The aim of the classification is to determine the class of the element. The authors have chosen gradient boosting algorithm for CatBoost library on Python to process raw data. Both this library and algorithm parameters have been empirically considered as the most efficient ones (Iterations: 3000, Depth: 4, L2_leaf_reg: 3, Learning_rate: 0.1).

The results of the experiment are listed in Table I.

TABLE I. OBTAINED PATTERN PARAMETERS

Parameter	Classifier value
Accuracy	92%
Macro Average Precision	92%
Macro Average Recall	96%
Macro Average F1-Score	94%
Weighted Average Precision	94%
Weighted Average F1-Score	92%

Along with the classifier, we have developed a special NLP (Natural Language Processing) module, which is able to identify the following text parameters in the scientific works [21]:

1) If there is any aim.
2) If there are any tasks.
3) If the work topic corresponds to the field.
4) If tasks and aim are stated in the introduction.
5) If the introduction corresponds to the stated topic.
6) If the relevance is stated in the introduction.
7) If the relevance is confirmed by the introduction.

IV. DEVELOPMENT PROCESS

The project is based on flexible project management methodology Scrum (https://www.scrum.org/), and open-source software code development (https://opensource.org/osd). The open-source development allows to involve developers and scientists all over the world. The authors have chosen GitHub as a repository (https://github.com/).

The repository allows collaboration, helps to make quick amendments and control team cooperation. If you want to assign a task or describe planned amendments, you should use the form called "Issues". The data in this form is used to create new lines, which further transform into pull-requests. One of the main peculiarities of the project is that the pull-requests should be reviewed at least by three team members. This permits each member of the team to get the idea of the project code base and consequently to reduce the possible code errors.

The authors have downloaded project documentation, which is compiled based on (https://docsify.js.org), so that the third parties can get an insight about the project architecture and required methods.

The project development formalization process should be carried out taking into account the chosen project management methods. This is of utmost importance at "the initial stage" [22]. This has been of great aid for the authors of this paper in the process of development

The repository and methodology, employed within the framework have made it possible for the authors of this paper to obtain the following positive results:

1) The number of code errors has been reduced.
2) The code has become more coherent (legible).
3) Now each team member has his own tasks and clearly understands what to do.
4) Each team member is free to choose one or another task to deal with.
5) Now each team member has a deeper insight into all code details.
6) The documentation is accessible.
7) The process of creating new tasks has become more flexible.
8) Any third party can collaborate and take part in the development process.

Moreover, if you use a remote repository, you can resort to automated code check for the compliance with the PEP8 standard. In addition, you will be able to perform tests after each iteration of changes.

The open-source code makes it possible for everyone to make his own contribution to the project development. The project can be found at https://github.com/normcontrol.

Project development and project management are simultaneous and interrelated processes. Each of them has its own standards, GOSTs, and operating regulations. Our project has two lines: product development and project management (Fig. 1) to boost general and individual performance and obtain more positive results.

The data formalization process, specified in standards, GOSTs and work regulations is based on finite state machines theory [23] regarding**:**

1) **Project development process,** based on GOST 7.32-2017, GitHub development regulations, open-source concepts.
2) **Project management process,** based on flexible project management methods Scrum, industrial professional standards [24] that are in effect in the Russian Federation.

When you are going to identify, assign or perform any project tasks or determine and complete any project stages [25], you should formalize structural elements according to the rules, stated in the official requirements, specified by standards, GOSTs and work regulations. Therefore, the process of checking structural elements for compliance with the rules is of a kind of an algorithm, which is based on the requirements, stated in standards, GOSTs and work regulations.

True assertions can form "a logic pattern" and "a mathematical pattern" [26]. The data can be actualized in "an object pattern" [27]. This is true for both development and management processes. It may help to implement automation, but we should call for skilled software engineers.

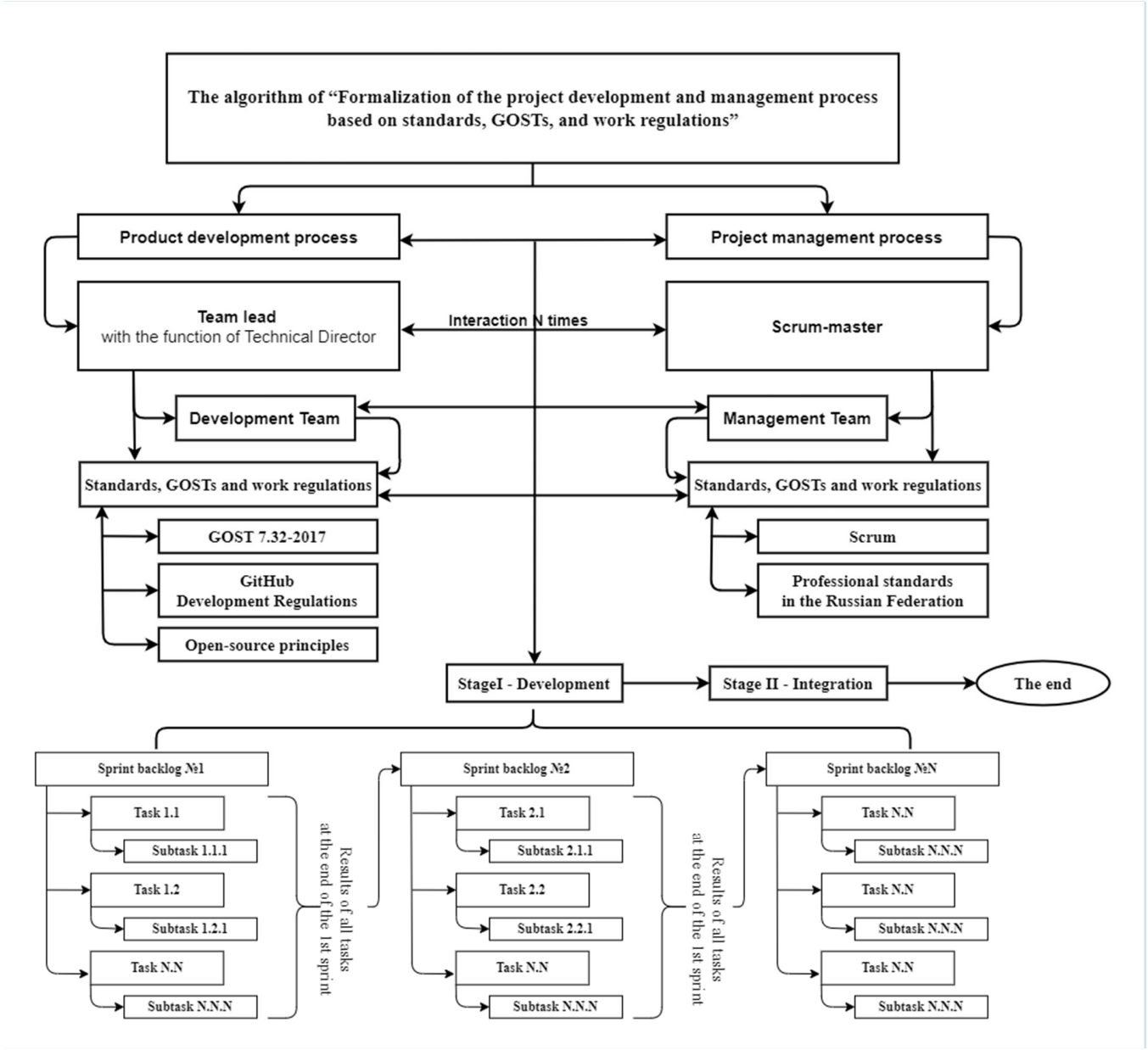

Fig. 1. The process of "Formalization of the project development and management process based on standards, GOSTs, and work regulations"

At "the initial stage" of the project, when we deal with "the project design" [22], it is crucial to fulfill the following tasks to ensure further timely performance:

1) To find the standards, GOSTs and work regulations required for each process.
2) To identify the main tasks for the development process and correlate them with the management process.
3) When two processes interact with each other, the big tasks should be divided into smaller ones. Then we can start working with these tasks on schedule, maintaining the interaction between the processes as much as it suffices, until the project finishes.

V. SERVICE STRUCTURES OVERVIEW

The aspects mentioned above have defined the structure of "Web tool for automated document formatting verification" (Fig. 2). This structure includes:

1) "Document parser" module performs parsing of ODT, DOCX and PDF. All the structural elements and their properties have been unified irrespective of the file type. The unified structure has been defined in the data class set named "document class".

2) "Flask API" module controls the interaction between other modules and performs documents compilation formatting verification.

3) "Django Application" web service provides an interface for user-library interaction. This interface comprises the following options: login, registration and user personal account; file download for review, errors statistics, based on review results, checking rules development.

4) "Classification" module contains a smart pattern, which is able to determine the class of any structural element in the document, even if the formatting of this structural element is wrong. The pattern uses the content and properties of a structural element to identify its class.

5) In addition, a "NLP" module checks the paper for tasks and aim, ensures that the paper title corresponds to its content and verifies other aspects, related to text analysis.

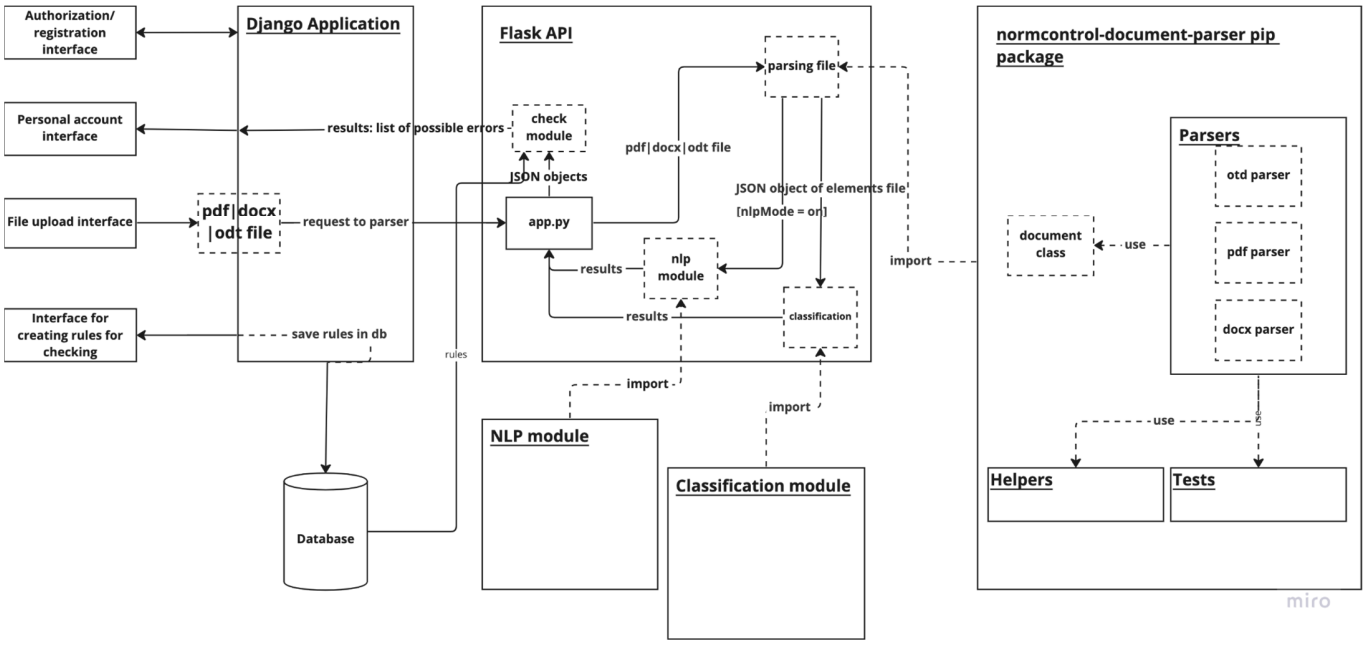

Fig. 2. Structure of document formatting verification service

Module "normocontrol-document-parser" is presented as a package for PyPi. That means it can be used independently of document formatting verification service. The module also contains tests, supplementary classes (for example, for units conversion) and documents.

This structure allows to make swift changes, without interrupting service functioning. All this results in flexible development process and proper system functioning.

VI. Results

The authors of this paper have developed a special tool and a library in order to check, if the formatting rules for the various kinds of documents are met. The document types under review are summary, term thesis, report on results of the research paper, graduation thesis, master or postgraduate dissertations.

The project consists of several parts: parsing modules for ODT, DOCX, PDF documents, a set of classes, aimed at unification the properties and content data of structural elements of the document; element classifier; RESTful API based on Flask; a NLP module (Natural Language Processing) for text analysis; user web interface.

Parsing modules (https://github.com/normcontrol/normcontrol-Document-Parser) are responsible for the extraction of text, images, tables, formulas, and their properties. Then these modules unify and standardize the data according to the rules for the corresponding structural element class.

Element classifier (https://github.com/normcontrol/catBoostClassifier) – is a smart pattern (CatBoost) that analyses one of the document objects and then determines the class to which it belongs.

NLP module (https://github.com/normcontrol/nlp-docx) is used for text analysis and helps to resolve the following issues: to identify, if there is an introduction and table of contents in the document; to determine, if the introduction corresponds to the further content; to predict to which field the work belongs to, to state obligatory elements for works.

The database contains reference properties for formatting of report structural elements. That permits to perform check employing various formatting standards.

As a result, we can conclude that the algorithm "Formalization of the development and project management based on standards, GOSTs and work regulations":

1) Has been successfully implemented in project management and document formatting verification.

2) The success is explained by cross – scientific and inter disciplinary approaches.

VII. Discussion

In their further research works the authors plan to improve parsing methods, extend extraction list and add neural network to project architecture. It will help to verify the accuracy of

formatting for elements that are difficult to check by traditional algorithms, formulae, for example.

It is worth mentioning that nowadays a web tool performs the documents formatting check per standards, stipulated in GOST 7.32-2017. However, if we make some amendments to code, it will be possible perform document formatting verification using standards stated in other regulatory documents.

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

The authors of this paper have developed a complex project, targeted at document formatting verification automation. This project comprises the set of classes for document properties storage, the element classifier, the NLP module for text analysis, the web tool, and the checking rules database.

Our web tool ensures proper formatting of articles and graduation theses in compliance with the requirements of periodicals, GOSTs, local acts; helps to implement asynchronous learning and customize the documentation preparation courses.

The authors have elaborated RESTful API at Flask framework to integrate all tools. It acts as a mediator between parsing modules, the element classifier, the NLP module, the database, and the web tool.

The next iterations give us the opportunity both to find mistakes for certain types of documents and correct them automatically [28].

The relevance of the paper:

1) The time, spent on document formatting verification can be significantly reduced. Document formatting verification can be automated.

2) More elements can be checked.

3) The authors of this paper have created a library that can help in extraction of structural elements and their properties from DOCX, ODT and PDF documents.

4) The authors of this paper have proposed their approach to the development of the software for scientific projects.

In further works, associated with the project, the authors are going to conduct an educational experiment and improve the algorithm “Formalization of the project development and management process based on standards, GOSTs and work regulations”.

REFERENCES

[1] A.A. Sokolov, A.M. Dvoryankin, and A.Yu. Uzhva, “The Development of the method of automation of the process of normative control of technical documentation”, *Proceedings of the Volgograd State Technical University*, No. 22(125), 2013, pp. 113-118.

[2] A.A. Nazartseva, and A.V. Gubarev, “Improving the process of standard control of technical documentation”, *Product quality: control, management, improvement, planning: collection of scientific papers of the 5th International Youth Scientific and Practical Conference*, Kursk: Closed Joint Stock Company "University Book", 14 Nov. 2018, pp. 84-87, eLIBRARY ID: 36659201.

[3] E.A. Trifonov, A.I. Soglaeva, and O.V. Anikeeva, “Optimization of standard control of technical documentation of the enterprise”, *Quality in production and socio-economic systems: collection of scientific articles of the 10th International Scientific and Technical Conference*, Kursk: South-West State University, 15 Apr. 2022, pp. 393-396, eLIBRARY ID: 48362100.

[4] S.D. Blinova, and A.A. Permyakova, “Automation of standard control of final qualifying works”, *Society, science, innovations (NPK - 2014): All-Russian annual scientific and practical conference: collection of materials: university-wide section, BF, GF, FEM, FAVT, FAM, FPMT, FSA, HF, ETF*, Kirov: Vyatka State University, 15-26 Apr. 2014, pp. 938-942, eLIBRARY ID: 23945230.

[5] Nail Nasyrov, Mikhail Komarov, Petr Tartynskikh, Nataliya Gorlushkina, “Automated formatting verification technique of paperwork based on the gradient boosting on decision trees”, *Procedia Computer Scienc*e, Vol. 178, 2020, Pages 365-374, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.11.038.

[6] StyleEase Software official website, CNET Download, Web: https://download.cnet.com/developer/styleease-software/i-10111284/.

[7] B.E. Starichenko, and M.A. Ustinov, “The program of automation verification of text documents formatting”, *Pedagogical Education in Russia*, Nol. 8, 2018, pp. 163-168.

[8] T.J. Sefara , M.S. Mbooi, K.J. Mashile, T.Rambuda, and M.R. Rangata, “A toolkit for text extraction and analysis for natural language processing tasks”, *2022 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, Durban, South Africa, 4-5 Aug. 2022, pp. 1-6.

[9] J. Hu, Y. Liu, “Analysis of Documents Born Digital”, *Handbook of Document Image Processing and Recognition,* Springer-Verlag, Springer, London, 2014, pp. 775–804, doi: 10.1007/978-0-85729-859-1_26.

[10] G. Endignoux, O. Levillain, and J.Y. Migeon, “Caradoc: A Pragmatic Approach to PDF Parsing and Validation”, *2016 IEEE Security and Privacy Workshops (SPW)*, San Jose, CA, USA, 2016, pp. 126-139, doi: 10.1109/SPW.2016.39.

[11] V.I. Martsinkevich, G.S. Larionova, V.V. Tereshchenko, K.A. Sitnikova, and N.N. Gorlushkina "Analysis of the possibilities of parsing electronic text documents for the automation of standard control", *Economics. Right. Innovations*, No. 3, 2022, pp. 39-49, (In Russ.), eLIBRARY ID: 49740779, doi: 10.17586/2713-1874-2022-3-47-57.

[12] K. Husain, “Extract and parse ODF files with python”, *Linux Journal*, 1 Apr. 2007.

[13] I.E.S. Rønningen, ”Standards - access for everyone. Converting OOXML, ODF and HTML”, Master thesis, University of Oslo, 3 Aug. 2009.

[14] P.V. Zahorodko, and P.V. Merzlykin, “An approach for processing and document flow automation for Microsoft Word and LibreOffice Writer file formats”, *Proceedings of the 4th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW 2021)*, Virtual Event, Kryvyi Rih, Ukraine, 18 Dec. 2021, pp. 66-82.

[15] J. Kong, Q. Zhou, T. Ma, Y. He, and H. J. Kong, "Word File Parsing Based On Python", *2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, IEEE, Shanghai, China, 2021, pp. 553-556, doi: 10.1109/AINIT54228.2021.00113.

[16] A. Kulkarn and A. Shivananda, *Natural Language Processing Recipes. Unlocking Text Data with Machine Learning and Deep Learning using Python,* Apress Berkeley, CA, 29 Jan. 2019, https://doi.org/10.1007/978-1-4842-4267-4_1.

[17] N.F. Nasyrov, D.M. Lushnikov, P.S.Tartynskikh, A.V. Berezhkov, “Application of approaches to optimizing the architecture of information systems using user feedback in the context of compliance with the requirements of the TOGAF standard”, *Modern science: actual problems of theory and practice. Series: Natural and Technical Sciences*, No. 10, 2021, pp. 89-95, (In Russ.), eLIBRARY ID: 47457112, doi: 10.37882/2223-2966.2021.10.14.

[18] L. Hao, L. Gao, X. Yi, and Z. Tang, "A Table Detection Method for PDF Documents Based on Convolutional Neural Networks", *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, Santorini, Greece, 2016, pp. 287-292, doi: 10.1109/DAS.2016.23.

[19] M. Shao, R.P. Futrelle, “Recognition and Classification of Figures in PDF Documents”, *IAPR International Workshop on Graphics Recognition, Ten Years Review and Future Perspectives*, GREC 2005, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 3926, pp. 231–242, https://doi.org/10.1007/11767978_21.

[20] S.Y. Kahu, "Figure extraction from scanned electronic theses and dissertations", *Masters Theses*, Virginia Tech, 29 Aug. 2020, http://hdl.handle.net/10919/100113.

[21] A. Kozyreva, U. Nazarenko, A. Berezhkov, and N. Nasyrov, "Analysis of the text of the FQP for automated standard control of documents", *In Journal of Physics: Conference Series*, IOP Publishing, vol. 2131, No. 2, p. 022102, Dec. 2021, doi: 10.1088/1742-6596/2131/2/022102.

[22] E.A. Kobets, N.N. Gorlushkina, "The Review of management methods and tools in social and economic systems and the Analysis of their application in project management", Audit and financial analysis, No. 1, 2021, pp. 176-186, (In Russ.), doi: 10.38097/AFA.2021.17.84.028.

[23] N.I. Polikarpova and A.A Shalyto, *Automata programming*, Saint-Petersburg: 2008, https://is.ifmo.ru/books/_book.pdf.

[24] Register of Professional Standards RosMinTrud official website, Web: https://profstandart.rosmintrud.ru/obshchiy-informatsionnyy-blok/natsionalnyy-reestr-professionalnykh-standartov/reestr-professionalnykh-standartov/.

[25] S.A. Gavrilov, A.S. Kyzdarbekova, S.S. Reznikov, E.A. Kobets, and N.N. Gorlushkina, "Modelling a project for the development of a neurocomputer interface based on EMG", *IOP Conference Series: Materials Science and Engineering*, vol. 940, No. 1, 2020, pp. 012018. DOI https://doi.org/10.1088/1757-899X/940/1/012018.

[26] E. Kobets, A. Tretiakov, and N. Gorlushkina, "Creation of Logical Models for Conducting Forensic Linguistic Expertise", *Advances in Social Science, Education and Humanities Research*, vol. 646, 2022 pp. 188-192, doi: 10.2991/assehr.k.220208.033.

[27] E.A. Kobets, N.F. Nasyrov, P.S. Tartynskikh, and N.N. Gorlushkina, "Creation the object model of the text benchmark document for automation the norm rule checking service", *Modern Science: actual problems of theory and practice. Series: Natural and Technical Sciences*, No. 8, 2021, pp. 57-63, eLIBRARY ID: 46716043, doi: 10.37882/2223-2966.2021.08.14.

[28] V. Venkatesan, and G. Hariharan, "An Implementation Approach Towards The Automation Of Document Formatting Using Python", *International Journal of Aquatic Science*, vol. 12, Issue 02, No. 2, Jun. 2021, pp. 3946-3959.