Automated Text Document Compliance Assessment System

Maria A. Zhigalova
Department of Information Technologies in Business
National Research University Higher School of Economics
Perm, Russia
mariezhigalova@gmail.com

Alexander O. Sukhov
Department of Information Technologies in Business
National Research University Higher School of Economics
Perm, Russia
ASuhov@hse.ru

***Abstract*—The study is dedicated to the problem of automating an electronic text document compliance assessment in accordance with the formal requirements on formatting set in standards. The need for the software system development of such kind appeared due to laboriousness and inefficiency of manual text check. The system functionality is based on the application of the Open XML SDK solution with the use of FormattingAssembler module included in PowerTools for Open XML. The system provides a comprehensive text document check in accordance with the formatting parameters defined by the user. In practice, the software product can be used to verify compliance with the formal requirements of research papers and dissertations, scientific publications, technical documentation, etc.**

Keywords—formatting rules, text document, compliance assessment, DSL

I. INTRODUCTION

Text processing, which refers to automation of creation and manipulation of electronic texts, has always been one of the primary disciplines in computer science. It involves determining the quality of publications, identification of potential duplication, plagiarism, partial borrowings, classification and clustering of documents, formation of databases and extensive collections of texts. Despite the fact that document checks in accordance with formatting rules does not imply detailed text processing, it should be noted that this procedure in one way or another is related to general text analysis and has its specific features.

It is known that document checks in accordance with formatting rules is primarily manual. It is considered to be extremely laborious and time-consuming, and researchers, as well as individuals responsible for document check in universities or organizations, are likely to appreciate the simplification of this process. Since the structure of research papers, dissertations, scientific publications, technical documents, etc. is a standard-based compulsory requirement, there is an ongoing need in the instrument allowing users to check the formatting of their work and automatically fix it if necessary. Therefore, the goal is to provide users with such functionality by creating a relevant software product minimizing the time and effort required.

Thus, the focus of the study is on the development of the automated text document compliance assessment system. It is assumed that the software product functionality is extended to the check of such formatting characteristics as page layout settings, styles parameters, headers and footers properties etc. In other words, the document design (not its content) is to be checked. The application can be used by a wide range of users, including students, teachers, technical writers, etc. It is expected, that the automation of text document compliance assessment will significantly increase the efficiency of business processes connected with document check.

II. RELATED WORKS

To date, there are two basic methods of control of the text on the absence of formatting errors and verification of a document in accordance with certain standards including ready-made design templates and various software solutions.

Violation of document styling often occurs when text is copied into a document from sources with diverse formatting patterns. Although this issue can be partially solved by application of built-in styles, the probability of error still exists. In this case, the use of formatting templates is a reasonable option.

One of the means of creating such templates is a markup language DocBook, which is an application of XML/SGML (XML – eXtensible Markup Language, SGML – Standard Generalized Markup Language). It provides a user with a unified set of tags for setting formatting of a text document [1]. This approach makes it possible to isolate document content from its style representation. The apparent advantage of DocBook is that a predefined set of tags eliminates formatting errors and allows a large number of users to work with the same text simultaneously.

Formatting templates are also utilized by the LaTeX publishing system which provides the capability for automating a process of inputting and formatting text of a document. The content of a LaTeX document, similarly to DocBook, is represented by structural and semantic markup. Text document formatting is described in a separate file with style information [2] which defines formatting rules, specific to each document type. Despite a vast variety of functional characteristics, it should be mentioned that LaTeX has a

number of disadvantages: firstly, in order to manipulate LaTeX documents, it is required to have a special development environment installed on a computer, and secondly, the process of creating a LaTeX document may be challenging for users who are not sufficiently skilled to work with the LaTeX system.

The automation of compliance assessment is implemented in a number of software products, one of which is an intelligent web-based system for spell checking "Orogrammka". The software checks the norms of grammar, punctuation and document formatting [3]. Compliance assessment is provided for research papers and dissertations in accordance with requirements that are set in a number of standards supported by the service. The software has an intuitive and simple interface, however, it should be noted that text check is limited to a strictly predefined set of formatting rules (margin sizes, page layout settings, reference list format, etc.) without the possibility of expanding the functionality by a user.

Another tool for automated formatting rules check was developed in Volgograd State Technical University [4]. This software solution is a Microsoft Word 2007 add-in which allows users to check their documents and fix detected errors. In spite of convenience and ease of use, the service has a significant drawback: users whose personal computers are not running Microsoft Office Word are deprived of the opportunity to perform the compliance assessment of text documents.

Overall, the analysis of the studies mentioned above highlights the need for the software system that provides an extensive functionality for formatting rules check, yet, has a user-friendly interface appealing for a large group of users. This work is to propose such a system.

III. TEXT DOCUMENT FORMATTING

A. Overview and Comparative Analysis of Popular Text Document File Formats

It is known that electronic text documents represent a major part of stored and processed data. This explains availability of a significant number of file formats used for specification of textual information. However, due to the fact that formatting check of various types of text documents requires the use of special software tools, there is a need for selecting the most appropriate file format which is to be used as the basis for the development of a software product.

Thus, the most common editable text file formats were identified:

- OpenDocument Text (*.odt) – a file format for text documents with an open specification standardised by ISO/IEC 26300; based on XML.
- Rich Text Format (*.rtf) – a closed cross-platform file format for storing text documents developed by Microsoft. A document with *.rtf extension consists of commands which can be divided into control words and control characters.
- Microsoft Word (*.doc) – a proprietary binary text file format used in Microsoft Word 97-2003. Document files represent complex objects organized according to the rules of structured storage [5]. The basic unit of data measurement is a symbol; all information about characters is in document stream.
- Microsoft Word (*.docx, *.docm) – an open file format for storing electronic text documents used in Microsoft Word since version 2007. DOCM extension indicates support of built-in macros and scripts. Microsoft Word with DOCX (DOCM) extension is part of the Office Open XML format. Office Open XML was initially standardized by Ecma-376 and then redefined in ISO/IEC 29500 standard [6]. OpenXML is a structured archived file that contains markup of a document in an XML format, graphical information and other data included in this text document.

Table I contains results of the comparison of electronic text documents formats by a number of parameters that will identify the option most preferred for research purposes.

TABLE I. TEXT DOCUMENTS FORMATS COMPARISON

	OpenDocument Text (*.odt)	**Microsoft Word**		**Rich Text Format (*.rtf)**
		***.doc**	***.docx (*docm)**	
Date of creation	2005	1997	2007	1982
Open or proprietary	Open	Proprietary	Open	Proprietary
Document file self-sufficiency	Partial	Full	Full	Partial
Ability to convert to other formats	Yes	Partial	Yes (partial for *.docm)	Yes
Free software	Yes	Partial	Partial	Yes
File size compactness	High	Low	High	Low

Unlike Rich Text Format and Microsoft Word (*.doc), OpenDocument Text and Microsoft Word (*.docx, *.docm) file formats have open specifications which allows third-party developers to freely create software for processing text documents with ODT and DOCX extensions. It is also worth mentioning that ZIP archive compression used by these formats significantly reduces file sizes making them more compact.

Microsoft Word documents of all versions are self-sufficient, i.e. they store all necessary data for correct content representation, whereas OpenDocument Text documents may not be displayed correctly in different programs or operating systems and RTF is fully supported only in a limited number of software products. The capability to convert from one format to the other is represented in every case. Comprehensive free software is only available for OpenDocument Text (some features of Rich Text Format are not implemented in freely distributed products). However, it is worth noting that, despite strong connectivity of Microsoft Word to the original Microsoft software and the absence of free alternatives, the usage of the format prevails. Such a conclusion can be drawn

on the basis of statistics from Microsoft [7], according to which about 1.2 billion people around the world use Microsoft Office applications as their primary tool when working with spreadsheets, texts, presentations, etc.

The advantages of a Microsoft Word file format based on an Open XML format [8] include:

1. Interoperability. The capacity of the format to interact and function with a large set of both custom and commercial applications provides a high degree of compatibility of documents for different tasks.

2. Backward compatibility. The ability of transformation of MS-DOC files into Open XML format with high accuracy allows end users to convert these documents to the Open XML format, and then programmatically access the converted documents.

3. Programmability. Minimum requirements for working with Open XML include a tool that can open and save ZIP files and an XML parser/processor. ZIP and XML libraries allow creating documents in Open XML format on a software level.

4. Integration of business data. Office applications support custom XML schemas that can extend the capabilities of the existing Office document types. Thus, users can export data from existing systems to the documents in the Office file formats.

5. Compact file format. Open XML format uses the technology of ZIP compression for storing documents which provide the possibility of reducing storage space. Opening the file causes the automatic unpacking of the archive, and saving the file results in its compressing.

Thus, a comparative analysis of the formats of text documents showed that Microsoft Word (since 2007 version) seems to be the most appropriate option in terms of the use of open standards based on ZIP and XML, the capability of processing in third-party applications, the ability to convert to other formats and popularity among users

B. WordprocessingML Description

An ISO/IEC 29500 standard specifies a markup language for text document description which is called WordprocessingML. In a WordprocessingML file elements are grouped in accordance with functionality and stored in separate parts of a ZIP archive. For example, information about all footnotes in a document is gathered in one element, however, in case of footers, the situation is slightly different: each section of the document can store up to three different configurations of headers and footers with different numbering options, special first page settings, etc. Thus, the structure of WordprocessingML includes a set of the following elements: a main document, comments, document settings, footnotes, header/footer, styles, fonts table, document glossary, etc. Fig. 1 illustrates parts of a document TestFile.docx opened with a tool Open XML Package Editor PowerTool for Visual Studio that allows to view the file hierarchy of the document archive and the relationships between them and also to modify their markup.

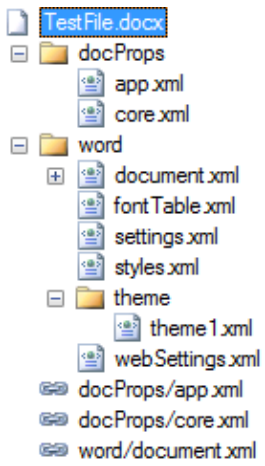

Fig. 1. Microsoft Word document file structure

In the main document part paragraphs (w:p) and tables (w:tbl) can be child elements for document body (w:body), table cell (w:tc) or text box (w:txbxContent). Paragraphs, in their turn, are a run-level content container for text runs (w:r), or images – a VML document (w:pict) or a DrawingML object (w:drawing). Finally, sub-run-level content incorporates multiple text elements (w:t).

Formatting of a text document with the use of Microsoft Word refers to implementation of various styles with parameters included in styles.xml file of a document archive [9]. This file contains data on styles of paragraphs, characters and tables, latent styles and standard settings of styles for an entire document (document defaults). Styles of paragraphs, characters and tables comprise information about current formatting of a document, whereas hidden styles are not used directly and serve primarily as a cache repository for style settings, for example, the ones copied from a template. Standard styles store default values for the entire document formatting. However, it should be noted that styles.xml file does not involve data on formatting of numbered and bulleted lists that is included in a special numbering.xml file.

The fact that content of a document can be formatted on multiple levels leads to a problem of determining a comprehensive set of formatting parameters used for a particular paragraph or a run of the text. These levels of formatting are schematically represented in Fig.2.

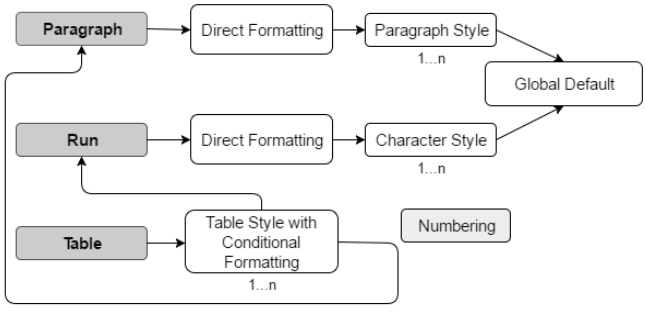

Fig. 2. Levels of Microsoft Word text document formatting

Thus, if it is needed to retrieve information about a paragraph (e.g. line spacing or indentation), the first aspect that has to be checked is direct formatting which is specified in a file called document.xml. Yet, paragraph parameters might not be indicated in this file, and, in this case, it is necessary to inspect the style which is referred to in paragraph properties. If this style does not contain data on the paragraph formatting, then the styles from which it inherits are to be checked. If this action did not bring any results, then the only option left is to process the contents of the node Global default, i.e. default settings of all styles in a document.

Similar approach is credible for checking text runs formatting (defining such font settings as size, name, etc.); the only difference is that character styles are put into consideration and they can also form an inheritance hierarchy.

Data on tables formatting are defined in styles with conditional formatting that specify the properties of rows and columns. Table styles are also inheritable. Text inside table cells is checked according to algorithms of determining formatting of paragraphs and runs. In case of numbering, each list item may include formatting from a paragraph, a numbering format in numbering.xml or a style that is indicated by this format.

Overall, the major difficulty of text document formatting check lies in determining precise formatting parameters for paragraphs, tables, numbering and runs of text for the purpose of conducting as extensive an analysis of conformity of a document to specified rules as possible.

IV. System Development

The compliance assessment procedure can be described as follows: the system sequentially retrieves formatting data from document markup and compares it to formatting parameters specified by the user. In order to work with WordprocessingML markup, it was decided to use Open XML SDK 2.5 for Microsoft Office. Retrieval of information on document formatting was performed by using the FormattingAssembler module which is a part of PowerTools for OpenXML. This module accesses style information on every level of formatting and assembles it, so that the markup of an original document is modified in a way that there is only direct formatting left. However, this direct formatting contains all formatting parameters (even from hidden styles) that were applied to a document.

OpenXML SDK built on the System.IO.Packaging API allows users to manipulate documents that adhere to the Office Open XML File Formats Specification, e.g. documents created with Microsoft Office applications. This package provides a set of strongly-typed classes to obtain data about the formatting of a document and makes it possible to modify an original document (for example, to add comments).

Despite the fact that .NET offers standard assemblies for working with Microsoft Office, the preference was given to OpenXML SDK. COM Interop (Component Object Model) provides access to Word objects (sections, paragraphs, tables, etc.) and has functionality for creating and editing documents, however, it does not support server-side automation and processes documents markedly slower than SDK.

The analysis of documents demanding certain formatting resulted in identification of a number of essential parameters for assessing the accuracy of text document formatting. Thus, the system is to perform compliance assessment according to these parameters:

1) page layout (page margins, paper format, orientation, columns, page numbers, header/footer settings);
2) paragraph (spacing, indentation, alignment);
3) font (size, name, color, toggle properties – bold, italics, underlined);
4) numbering and lists (level, numbering format, start value);
5) tables (vertical and horizontal text alignment, borders, cell margins, width, table header);
6) images (placement – anchor or inline, size).

The process of text documents check can be divided into several stages. Firstly, it is needed to define a set of rules according to which compliance assessment will be performed. The system provides a user with a possibility to specify design requirements for various documents by loading a formatting template or entering parameters manually, modify these requirements, or delete them if necessary; all information is stored in a formatting rules repository.

The second step is to upload the document into the system and select appropriate formatting rules. After that check of a document can be performed. The system reloads the document and adds comments with identified inconsistencies between the formatting used in a checked document and specified formatting requirements (see Fig. 3). So, in this case the system has detected that sizes of a header and a footer, page margin sizes, and some settings of a style "Heading 1" were selected incorrectly, and all this information was reported to the user. It should be noted that if there are no formatting mistakes in the original document, the system will not create any annotations. Comments on inaccurate paragraphs styling are added accordingly to each paragraph with incorrect formatting; notes on violation of formatting requirements for page layout settings, header/footer, etc. are added to the first paragraph of text. If there are formatting errors inside paragraphs or runs of text (for instance, some word has odd font settings), the system makes comments on each word in particular.

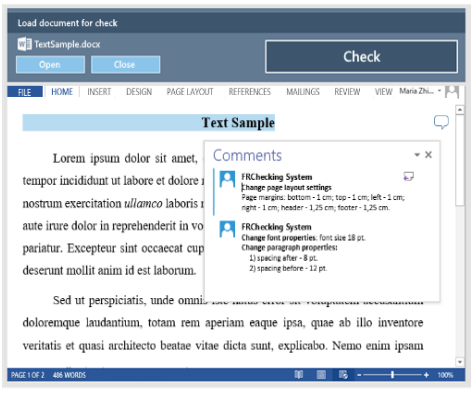

Fig. 3. Compliance assessment system interface

Thus, the user-system interaction complies with a number of different scenarios. The first scenario (see Fig. 4) implies that a user enters formatting rules manually, and then loads an original document for the check. In this case, the system (FRC System – Formatting Rules Check System) provides a user with either the resulting document containing the notes or the one with formatting corrected in accordance with rules specified by a user.

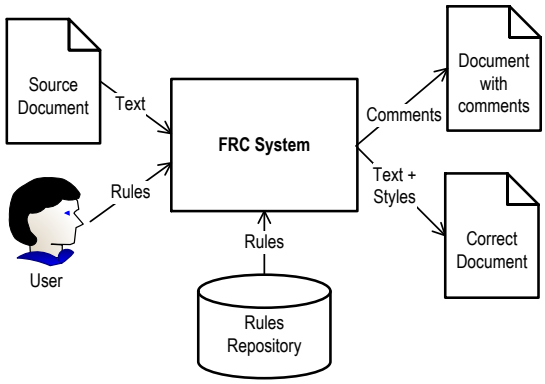

Fig. 4. Correct document generation

According to the second scenario (see Fig. 5) a user uploads a properly formatted template document, the system performs its analysis and downloads its formatting rules into the rules repository. This procedure significantly simplifies the entry of formatting rules of a document.

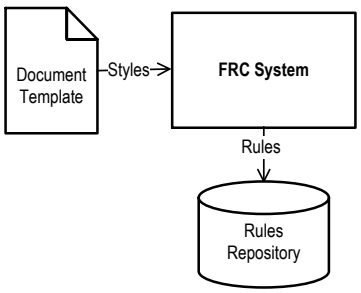

Fig. 5. Creation of rules based on document template

The third scenario of the interaction (see Fig. 6) suggests that a user manually enters formatting rules of a document, the system saves them in the repository, and then generates a document template with an automatically created styles which a user can use for further work with a document.

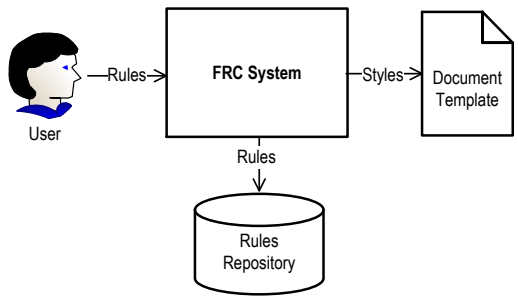

Fig. 6. Creation of document template based on rules

V. Text Document Structure Check

As noted earlier, the task of a text document analysis is not reduced to formatting rules check. In the more general case, it is necessary to analyze document structure, i.e. verify that all required sections are included. This problem often arises in preparation of design documentation, for example, in the process of developing information systems. Design documentation has a normative function, i.e. it contains mutual obligations of participants of a project that helps to avoid misunderstandings and abuses at the stage of handover-acceptance [10; 11].

The types and completeness of project documents are standardized. However, due to the fact that all technical documents are structurally very similar (they all consist of sections and subsections, may include additional documents, diagrams, tables, etc.), a special language for defining document structure and links between different documents can be developed. It will allow automating the process of analysis of an original set of project documents and generation of the new ones [12]. In the same way, it is reasonable to develop tools for extracting system requirements from the project documentation, and then control their compliance in the process of implementing the system. However, the process of creating design documents is quite a laborious task that requires precise knowledge of a document structure. This process can also be automated. Means of automating the generation of project documentation will allow generating a document template on the basis of descriptions of different sections of a document specified in a convenient visual user interface. This template can later be modified manually.

In order to describe documentation used in the process of information systems design, visual domain-specific language can be developed. Domain-Specific Language (DSL) is a modeling language designed for solving problems of a certain class in a particular domain. Unlike general-purpose modeling languages, DSL is more expressive, easy to use and intelligible to various categories of professionals, since it operates with the familiar terminology of the domain. Therefore, a large number of DSLs is designed nowadays in order to describe systems in different subject areas: artificial intelligence systems, distributed systems, mobile applications, real-time and embedded systems, simulation systems, etc.

Since description of project documents implies not only determining their structure, but also specifying the relations between them, the developed domain-specific language describing project documentation has two levels [13].

The first level of the language makes it possible to describe a set of documents and relations between them, the second level – the structure of a particular document. Due to a simple graphical notation of the language, the system can be used by IT-specialists, as well as clients who are not professional programmers.

VI. Conclusion

The main result of the work done is the developed system that automates the check of a text document in accordance with formatting rules specified by a user. As it was tested, the

system substantially reduces the complexity of operations performed and makes the process less time-consuming.

Moreover, the visual DSL for describing the structure of a document was created. This language can be integrated into the support system of work of an analyst when information systems are designed. On the one hand, this provides means to perform analysis and parsing of a set of design documents loaded into system, presenting the sections of a document as individual elements of a model. On the other hand, with the use of the developed language an analyst can describe each section of a design document separately, and then generate a single text description on their basis.

Despite the fact that the system performs all the main functions, there is still space for improvement. The system can be upgraded by developing web-interface for more convenient use and expanding the set of criteria for document check in order to perform more comprehensive compliance assessment.

REFERENCES

[1] S. Berdachuk, *Use the DocBook for Documentation Writing.* [http://www.berdaflex.com/ru/eclipse/books/rcp_filemanager/ch01s04.html] (Checked: 10.04.2016).

[2] S.M. Lvovsky, Typing and formatting in LaTeX System. Moscow: MTSNMO, 2006.

[3] Orfogrammka. Spelling Checking Web Service. [http://orfogrammka.ru] (Checked: 10.04.2016).

[4] A.A. Sokolov, A.M. Dvoryankin, A.Yu. Uzhva, "Development of the Method of Process of Technical Documentation Normative Control Automation," in Izvestia VSTU, 2013, no. 22 (125), pp. 116-117.

[5] K.E. Klementyev, *Internal MS WORD document format* [http://uinc.ru/articles/39/] (Checked: 10.04.2016).

[6] ISO/IEC 29500. Information technology – Document description and processing languages – Office Open XML File Formats. International Organization for Standardization, Geneva, Switzerland, 2012.

[7] Microsoft. Microsoft by the Numbers [http://news.microsoft.com/bythenumbers/planet-office] (Checked: 10.04.2016).

[8] OpenXMLDeveloper.org. Benefits of Open XML. [http://openxmldeveloper.org/wiki/w/wiki/benefits-of-open-xml.aspx] (Checked: 21.10.2015).

[9] W. Vugt, *Open XML Explained*. [http://openxmldeveloper.org/cfs-file.ashx/__key/communityserver-components-postattachments/00-00-00-19-70/Open-XML-Explained.pdf] (Checked: 21.10.2015).

[10] A.V. Zaboleeva-Zotova, Yu.A. Orlova, "Automation of Procedures of the Product Requirements Document Text Semantic Analysis," in Izvestia VSTU, 2007, no 3, vol. 9, pp. 52-55.

[11] Yu.A. Orlova, "Product Requirements Document Text Analysis Methods," in Izvestiya TSU. Engineering Sciences, 2011, no 3, pp. 213-220.

[12] M.A. Zhigalova, A.O. Sukhov, "Validation of the Design Documentation Based on Domain-specific Language," in Vestnik molodykh uchenykh PSNRU. Vol. 4. P. 224-228.

[13] M.A. Zhigalova, A.O. Sukhov, "Domain-specific Language for Describing Documents Used in Information Systems Design," in Izvestiya SFedU. Engineering Sciences, 2015, no. 2, pp. 126-134.