UDC 004.91

**THE PROBLEM OF SOFTWARE EDITING OF MICROSOFT WORD DOCUMENTS**

R. O. Popov, student

*Oles Honchar Dnipro National University*

N. V. Karpenko, Ph.D. of Phys.-Math. Sciences, Associate Professor

*Oles Honchar Dnipro National University*

**Keywords**: Microsoft Word, software document editing, document templates, document workflow automation, information technology.

Currently, there is a major problem in the Ukrainian education sector that is not directly related to teaching: the creation and processing of documentation that accompanies the educational process. Although information technology has made it easier to create documents in electronic format, it still takes a lot of time to fill out the paperwork. Difficulties arise when you need to create several documents for one object in a subject area, for example, different applications, questionnaires and contracts for one student for a competitive offer when applying to a higher education institution. Thus, the idea of developing software to automate these routine tasks using specialized libraries to organize document templates and fill them out.

Since the de facto standard for creating electronic documents in Ukraine is Microsoft Word, this paper will look at the available software editing tools for creating such documents and the specifics of creating software for processing such files.

Microsoft Word is a word processor developed by Microsoft and included in the Microsoft Office suite. Microsoft Word saves documents in the .docx format according to the Office Open XML specification [1]. Table 1 shows libraries for processing .docx files for different programming languages.

Taking into account these libraries, we can conclude that the document editing ecosystem is not developed and popular in all programming languages. Libraries with a full package of document editing capabilities are developed by commercial companies and are quite expensive to use. And beginner developers working with C# are forced to use the free version. And this is quite inconvenient, since, for example, GemBox.Document limits free use to 20 lines of text. Although sometimes free products are not inferior to commercial ones, as in Python (Spire.Doc and python-docx).

There is also the problem that libraries in one programming language may be more powerful than those in other languages. It is for these reasons that sometimes editing and generating .docx documents becomes an unpredictable problem. In some situations, the edited field has incorrect formatting, in others, the document stretches to more pages than allowed.

Table 1 - Libraries for processing .docx files

|  |  |  |  |
| --- | --- | --- | --- |
| Programming language | Library name | Terms of Usage | Development activity |
| C# | GemBox.Document | Limited free version and full paid version | Actively developed by GemBox |
| DocX | Limited free version and full paid version | Actively developed by Xceed |
| JavaScript | docx.js | Free | Developed by the community |
| Python | python-docx | Free | Recently, it has not been actively developed by the community |
| Spire.Doc | Paid | Developed by the company (has similar functionality to python-docx) |
| C++ | DuckX | Free | Recently, it has not been actively developed by the community |

The limitations of libraries also force you to reformat documents into another form that the library can understand. Another peculiarity of developing templates in .docx is that version control systems cannot identify specific changes in the file (changed lines), which slows down the software development process.

The reason for these disadvantages is that OOXML (the current .docx standard), although open, is very complex (unlike the already existing Open Document) [2]. It takes a lot of time and effort to develop a complete library, but only large corporations have the funding for such work. The fact that Word is primarily set up for WYSIWIG-style editing makes programmatic editing very difficult.

**List of references**

1. International Organization for Standardization Information technology — Open Document Format for Office Applications (OpenDocument) v1.2. 2015. URL: <https://www.iso.org/standard/66363.html>
2. Starynkevitch B. Comment of Basile Starynkevitch on StackOverflow question. 2018. URL: <https://opensource.stackexchange.com/a/6481>