



Research on the Design of Document Generation System Based on WEB

Shanshan Dong

School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China

Email: dong414@zjnu.edu.cn

How to cite this paper: Dong, S. (2026) Research on the Design of Document Generation System Based on WEB. *Open Access Library Journal*, **13**: e14865. <https://doi.org/10.4236/oalib.1114865>

Received: January 12, 2026

Accepted: January 31, 2026

Published: February 3, 2026

Copyright © 2026 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Traditional document generation systems often struggle to strike a balance between flexibility and usability: they either rely on rigid templates or require users to possess programming skills. To address this issue, this paper proposes a web-based document generation system built on a user-configurable template architecture. The main contribution of this work is the design and implementation of a web-based document generation system that integrates a graphical template configuration interface, enabling non-technical users to define document formats without programming. The system supports three main functionalities: 1) creating and managing customizable templates; 2) generating structured outline documents based on templates; and 3) automatically reviewing and correcting document formatting. Experimental results on a small user group indicate that, compared with manual formatting, the proposed system reduces document formatting time by approximately 45%.

Subject Areas

Artificial Intelligence

Keywords

Document Generation, Open XML, Django

1. Research Background

With the rapid development of modern technology and the widespread use of electronic office work, Word documents are used in the production and life of most enterprises and individuals. Each enterprise or individual has requirements for the document format when using Word documents, especially for some documents with specific purposes, such as government official documents, accounting files, and papers. Therefore, multiple similar operations are required to modify

and review the document format to ensure its standardization, which will surely consume extra time and manpower and affect the production efficiency of enterprises to a certain extent. In response to this, there is an urgent need in society for a document generation system that can achieve office document automation. Therefore, many document generation systems for specific fields or industries have emerged in today's society, such as graduation project document generation systems, automotive design document generation systems, and audit document generation systems. These documents have been successfully applied to various industries in society, and through research, they are found to have the following characteristics:

- 1) Spanning various industries, with a large market demand.
- 2) The application direction of the system has strong directivity and is not suitable for widespread use.
- 3) Lack of methods to meet personalized needs and being unable or inconvenient to update functions according to user requirements.

There are numerous enterprises in society, countless in various colors in China, with widely different fields, resulting in different requirements for document formats for each enterprise.

Even different departments within an enterprise have different format requirements. To better meet the needs of the majority of enterprises or individuals, users need a more personalized document generation system to meet their daily production and life needs, and this document generation system needs to be compatible with different scenarios and have strong adaptability.

Document generation software still has major problems today because it often requires developers to write code and some complex and lengthy iterations, which may be used to create and correct templates. Writing this part of the code takes up a lot of developer resources. If ordinary business personnel are asked to complete this part of the work, they often need to be trained to learn relevant knowledge, and it is easy to make mistakes during the writing process. [1]

In addition, document generation is closely related to document file formats. Closed office document formats have gradually become a shackle hindering user information exchange, restricting the vitality of documents, increasing the user's usage cost, and posing a security risk to users' data storage. Therefore, the trend towards open document formats has become an industry consensus [2]. Some organizations have established a basic consensus to use XML as the basis for future document file formats. Open standards based on XML (open format standards composed of XML) include DocBook, and more recently, the ISO/EC standard Open Document (ISO 26300:2006) and Office Open XML (ISO 29500:2008). [3]

2. Requirement Analysis

Since Hilton *et al.* proposed Deep Learning (DL) in 2006, deep learning-related technologies have achieved breakthroughs in many fields such as natural language processing, speech recognition, and computer vision. As an important research

area in natural language processing, text automatic generation has been deeply studied by many scholars, and many deep learning-based text automatic generation systems have emerged accordingly. [4]

In August 1991, the automotive design document generation system for the automotive industry began to be available to the public. At this time, the document generation system was used to replace manual work in sorting out automotive parts lists, and its main tasks included data statistics, sorting, summarization, and other management functions [5]. Subsequently, starting from 2000, document generation systems for specific fields such as database files and audit files gradually emerged. These document generation systems were highly targeted, with simple system functions, limited operability, and poor template extensibility, thus providing limited convenience to users [6]. Between 2002 and 2009, the number of research studies on document generation systems increased year by year and reached its peak. During this period, there were more and more XML-based document generation systems. As a markup language, XML has good scalability, flexibility, and self-descriptiveness, which can facilitate document operations to generate documents in a certain format. Additionally, in 2009, the document generation system based on the B/S architecture began to be available to the public [7]. The B/S architecture has some advantages that can better facilitate developers in their development work and can provide services to users anytime and anywhere. Therefore, web-based document generation systems can better serve the general public in society and be applied to the office fields of various industries. Since then, many document generation systems based on the B/S architecture have emerged, and there are also more and more document generation systems for specific fields. In 2020, the official document automatic generation system was made available to the public [8]. This system uses technologies such as natural language processing, machine learning, and big data analysis to automatically and intelligently generate specified documents dynamically. This system has a wide range of target users and strong comprehensive functions.

Generally speaking, today's document generation systems mainly use open standards based on XML. Among them, Office Open XML has many advantages. It has improved data transfer, information exchange, and interoperability with industry systems. Due to using the Office Open XML specification, file data can be exchanged more conveniently, improving the efficiency of developers and having a relatively good prospect [9]. Additionally, in the field of document generation, various technologies of artificial intelligence are constantly being applied. Among them, technologies such as natural language processing and machine learning are also constantly making breakthroughs, making document generation more automated, intelligent, and comprehensive [10]. Based on the combination of these two aspects, the main development trend of future document generation systems will be a trend based on XML and integrating artificial intelligence-related technologies.

This paper aims to explore and design a document generation system that can

meet the needs of the general public and has strong adaptability. This system can assist users in managing documents and provide functions for checking document formats, and can, to a certain extent, achieve automatic document generation and automatic document review, reducing the time and manpower spent by enterprises or individuals on office document formats and improving the office efficiency of users.

To complete the above system, the author's main research work includes: 1) Office document automation based on python-docx. 2) Use the Django framework to develop the website.

The proposed system is designed with the primary goal of improving user work efficiency. It adopts a Browser/Server (B/S) architecture, emphasizes interaction between users and the system, and enables users to access the system anytime and anywhere, thereby providing a superior interactive experience. The system is implemented using Python for the backend, HTML, CSS, and JavaScript for the frontend, MySQL as the database, Django as the overall framework, and Open XML parsing for document reading and writing. By integrating these technologies, the system supports web-based document generation and offers both personalized and general-purpose document processing functions. With this system, users can reduce the time, manpower, and financial resources spent on repeatedly adjusting document formats.

The main work of the system is to generate the outline document required by the user based on the title content given by the user and to modify the incorrect format of the user's document. In order to achieve the required effect, there must be an intermediary to specify the target document format. Therefore, in the system to be explored and designed, this intermediary is defined as a document template, which can be customized by the user or used with the templates provided by the system. To complete this system, the following functions need to be ensured:

1) Specify the document template format. Customize the format of the required document template according to user needs. This template will be used for subsequent document generation and format review. After use, the template will still be saved in the user data for subsequent users to use again, with good reusability. The system also provides users with a set of built-in templates or intelligent recommendations powered by large language models, classified according to different usage environments, such as notices, summaries, papers, etc., for users to search and use.

2) Generate the outline document required by the user. For this function, the user needs to provide the required title content and the number of title levels. The system will generate the required outline document according to the template used by the user. The outline document only includes headings at all levels, and users can directly enter content in the body part. The document generated according to the template will also be saved in the user data and will mark which template is selected.

3) Modify incorrect formats. When users want to review the format of the source document, they also need to provide a source document. The system modifies the incorrect format part according to the selected template and then provides it to the user.

The verification document generated according to the template will also be saved in the user data and will mark which template is selected.

3. Function Design

Based on user requirements, the system to be explored and designed divides the modules, mainly including data management, template management, document generation, format review, and administrator module.

Data management mainly saves the user's personal documents and personal template information. Users can view relevant information in the data management module. Documents can be downloaded and deleted, and templates can be edited and deleted. Template management includes creating personal templates and viewing system templates. Since the system to be explored and designed distinguishes between personal templates and system templates, personal templates are created by users, and the format can be defined by users themselves. System templates are given by the system, and users can only view the settings in them and cannot change them. Document generation mainly generates an outline document according to the template and the title content given by the user. In this module, users need to select whether to use a personal template or a system template, and then enter the title content and select the corresponding title technology in the graphical page to generate the document. In the format review module, users can upload local documents and then review the document format according to the selected personal template or system template. The places where the document format does not conform to the template settings will be modified, and the document with the correct format will be provided to the user after the modification is completed. The administrator module mainly operates on system templates. Administrators can enter the administrator interface to create system templates and change system templates. When creating system templates, administrators need to provide template types for users to choose from. When changing system templates, administrators can choose to edit or delete.

The system is specifically divided into five modules on the server side, namely the data management module, the template management module, the document generation module, the format check module, and the administrator module. The main functions are as follows.

3.1. Specify the Document Template Format

This function is the most important aspect of the interaction between users and the system, which is related to whether the user can use it conveniently. Users can set the desired document format through the graphical interface. If users do not set some formats, the system will set them according to the default format. After

setting, the template can be saved in the user data, and users can reuse the template later. In addition, the system provides a set of built-in templates and supports intelligent recommendations by large language models based on user-input keywords. The inclusion of system templates can reduce the time users spend on manually configuring templates. At present, our templates mainly include font family, font size, alignment, and line spacing for headings at different levels. To avoid redundant storage of style fields across different text hierarchy levels, this paper adopts a master–detail database design that decouples the modeling of basic template information from text style configurations. The template table is used to store template metadata, while the text style table uniformly describes the style attributes of body text and headings at all levels using a “template ID + text level” scheme. This design improves the normalization and scalability of the database structure. The specific design is shown in **Table 1** and **Table 2**.

Table 1. Template information table.

Field Name	Type	Description
template_id	int	Unique identifier of the template
title	varchar	Name of the template
username	varchar	Creator of the template
create_time	datetime	Creation time of the template

Table 2. Template text style configuration table.

Field Name	Type	Description
style_id	int	Unique identifier of the style record
template_id	int	Associated template ID (foreign key)
level	int	Text level (0 = normal text, 1 – 4 = heading levels)
font_id	int	Font type (foreign key)
size_id	int	Font size (foreign key)
is_underline	boolean	Whether the text is underlined
is_italic	boolean	Whether the text is italic
is_bold	boolean	Whether the text is bold
alignment_id	int	Alignment format (foreign key)
spacing_id	int	Line spacing (foreign key)

The complexity of this function lies in the fact that there are bound to be a lot of things to set in the document format. To meet people’s needs comprehensively, it is necessary to conduct multiple tests to discover the oversights in the system. Moreover, when saving template data to the database, the size of the table also needs to be considered.

3.2. Generate the Outline Document Required by Users

This function is the most crucial part of the system to be explored and designed,

which is related to whether the correct format of the document can be generated. When the document template is completed, the format of the target document has been specified, which provides a basis for generating the outline document format required by users. Before generating the document, the template must be selected first. The template determines the format of the final generated document, and then the target outline document is generated according to the content of each level of headings input by the user.

The document template is saved in a specific table in the database. When generating the document, the system will read multiple values in the template and modify the format in the source document according to these values. This operation necessarily requires multiple judgments and loops to achieve. For example, the system reads the font size of the first-level heading of a template in the table as 12 pt.

3.3. Modify Incorrect Formats

Due to the existence of the template, not only can the document be generated according to the template for the user's needs, but the format of the document can also be reviewed. If the user wants to review whether the title and body formats of the document are correct, this function will provide the user with a format review function. It will modify the user's source document according to the selected template and provide the user with a document with correct format after the modification.

4. User Study

To verify the practical effectiveness of the system in improving document processing efficiency, this study designed and conducted a user experiment. A controlled experimental method was adopted, in which 30 office workers with daily document processing experience were recruited and randomly divided into a control group and an experimental group. The control group completed document formatting tasks manually using Microsoft Word 2019, while the experimental group completed the same tasks using the proposed web-based document generation system after receiving brief operational instructions, which included a brief demonstration of the system interface and the functionality of its main features. The experimental tasks were divided into three levels according to processing complexity. Specifically, the basic formatting adjustment task required participants to unify the formatting of a two-page document by adjusting the font type and size of second-level headings and body text according to a given template. The multi-level heading outline generation task required participants to generate a hierarchical outline document of approximately one page based on a provided formatting specification and to correctly apply multi-level heading styles. The complex document format inspection and correction task required participants to comprehensively revise a six-page document containing multiple formatting errors, including adjusting the font type and size of fourth-level headings and body

text according to the prescribed guidelines. During the experiment, the time required for each participant to complete the tasks was recorded as an objective measure of document processing efficiency. In addition, after completing the tasks, participants' subjective evaluations of system efficiency and usability were collected through a structured questionnaire. The questionnaire employed a five-point Likert scale for quantification.

The experimental results show that the experimental group achieved shorter average completion times than the control group across all three task categories, with an average time reduction of 44.64%, and a greater advantage in high-complexity tasks. Statistical analysis using two-sided Welch's t-tests revealed that the differences between the two groups were statistically significant across all tasks ($p = 0.0106$ for the basic task, $p < 0.01$ for the medium-complexity task, and $p < 0.001$ for the high-complexity task). Questionnaire results further indicate that 93.3% participants in the experimental group believed the system reduced document formatting time and gave high ratings to the graphical template configuration approach. These findings confirm the practicality and efficiency advantages of the proposed system in real-world document processing scenarios.

5. Summary and Outlook

This paper mainly explores how to reduce the time cost users spend on document formatting and ultimately proposes a web-based document generation system. This system realizes the core functions such as customizing and managing document templates, generating outline documents based on templates, and automatically checking and modifying document formats, solving the problems such as difficult format specification, much repetitive labor, and poor template extensibility in traditional document processing, and providing users with a convenient and efficient document processing tool.

Looking ahead, there is still considerable room for optimization and expansion of the system. At the functional level, the intelligence of the templates can be further enhanced. For example, natural language processing technology can be introduced to automatically recommend matching templates based on the text content input by users, or intelligently adjust the default settings of templates based on users' historical usage habits. At the same time, the ability to automatically generate document content can be explored. Combining deep learning models, the main text content can be automatically filled according to the keywords or outlines provided by users, improving the automation and intelligence level of document generation. Through continuous technological iteration and function improvement, this system is expected to play a greater role in the field of office automation and provide users with a more comprehensive and intelligent document processing solution.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Chan, D.K.C. (1998) A Document-Driven Approach to Database Report Generation. *Proceedings Ninth International Workshop on Database and Expert Systems Applications (Cat. No.98EX130)*, Vienna, 26-28 August 1998, 925-930.
<https://doi.org/10.1109/dexa.1998.707516>
- [2] Wu, Q., Li, N. and Fang, C.Y. (2009) A Comparative Study and Conversion Between 'Biaowentong' and OOXML Text Processing Document Formats. *Computer Applications and Research*, **26**, 591-594.
- [3] Blind, K. (2011) An Economic Analysis of Standards Competition: The Example of the ISO ODF and OOXML Standards. *Telecommunications Policy*, **35**, 373-381.
<https://doi.org/10.1016/j.telpol.2011.02.007>
- [4] Pang, S.S. (2019) Research and Implementation of Bait Document Generation Based on LeakGAN. Master's Thesis, Beijing Jiaotong University.
- [5] Lin, Y. (1991) A System for Generating Automotive Design Documents. *Traffic and Computer*, No. 4, 14-19.
- [6] Pan, W.D. (2002) Audit Document Generation System. *China Audit Information and Methods*, No. 8, 7-8.
- [7] Song, S.P. (2009) Design and Development of Document Generation Based on B/S. *Computer Programming Techniques and Maintenance*, No. 20, 38-39.
- [8] Wang, B.B. (2019) Design and Implementation of an Automated Official Document Generation System. Master's Thesis, Dalian University of Technology.
- [9] Cheng, R. (2013) Tools for Creating and Modifying docx Documents Based on OpenXML. Master's Thesis, Dalian University of Technology.
- [10] Wang, X.M. (2019) Research on the Application of Natural Language Processing Technology in Project Document Management. Master's Thesis, Beijing University of Posts and Telecommunications.