Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
Информатики и радиоэлектроники
Кафедра инженерной психологии и эргономики

УДК

Рубанова
Ирина Александровна

ФОРМАТИРОВАНИЕ ПРОИЗВОДСТВЕННОЙ ТЕКСТОВОЙ ДОКУМЕНТАЦИИ

АВТОРЕФЕРАТ

на соискание академической степени магистра техники и технологии

по специальности 1-59 81 01 Управление безопасностью производственных процессов

И.А. Рубанова

Заведующий кафедрой ИПиЭ
кандидат технических наук,
доцент К.Д. Яшин

Научный руководитель
кандидат технических наук,
доцент В.С. Осипович

Нормоконтролер
ассистент Е.С. Иванова

Минск 2016

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Основной целью данной магистерской диссертации является разработка сервиса по автоматизированному форматированию текстовой производственной документации. В работе рассматриваются вопросы проектирования подобного сервиса.

В первой главе производится анализ аналагов и подобных систем, специальных программ для форматирования текста. Рассматриваются и анализируются методы и условия выполнения форматирования, для достижения максимальной эффективности и результативности. LaTeX – наиболее популярный набор макрорасширений (или макропакет) системы компьютерной вёрстки TeX, который облегчает набор сложных документов [6]. Благодаря различным упрощениям, использование макропакетов зачастую позволяет избежать изощрённого программирования. Распространный и эффективный продукт, однако в ней существует ряд неразрешенных вопросов, например, что продукт не являет программой типа WISIWIG. Но главное неудобство составляет то, что программа предполагает изначальное задание параметров, а не форматирование уже набранного текста.

В результате анализа аналогов были сделаны выводы и поставлены задачи диссертиции. Так же было сформулированно техническое задание на разработку.

Во второй главе анализируются функции системы и определяется структура. Рассматривается распределение задач между техникой и человеком в работе системы. Разрабатываются алгоритмы работы пользователей в соответствии с эргономическими требованиями.

В третьей главе проводится обоснование и выбор инструментов разработки. В результате вышеописанного выстраивается структура приложения. Основная задача системы по форматированию текстовой документации – распознать структурные элементы текста. Алгоритм распознавания так же представлен в третьей главе.

ВВЕДЕНИЕ

При наборе различной документации, докладных записок, технических спецификаций и тому подобного получается объемный и сложный документ со сносками, ссылками, рисунками, таблицами и формулами. Каждый документ должен иметь определенный вид и соответствовать определенному набору принятых стандартов. Форматирование – это изменение внешнего вида документа и его отдельных частей с целью придания ему лучшего восприятия и удобочитаемости. По сути, операции форматирования не изменяют смысла текстового документа, но улучшают его внешний вид. К операциям форматирования относят различные способы выделения текста, а именно: изменение параметров отдельных слов и словосочетаний; изменение параметров отдельных абзацев; оформление заголовков и подзаголовков; преобразование текста в список; преобразование текста в табличный вид; вставка автоматически создаваемых полей (номеров страниц, таблиц, рисунков и пр.).

При форматировании документа автор сначала решает, какие части текста и как он будет выделять, а затем пользователь вновь перечитывает текст, но уже для того, чтобы оформить его. Очень часто случается так, что выполненное форматирование необходимо изменить. Например, задать другие значения параметров символа или параметров абзаца и пр. Тогда процесс форматирования приходится полностью или частично выполнять заново. Очень многие пользователи выполняют форматирование вручную, а именно, применяют основной прием форматирования «выдели объект и установи для него новые значения параметров». Это занимает много времени, особенно если документ большой и в нем присутствует много элементов форматирования. И здесь большую роль играет человеческий фактор.

Однако структурирование и единство оформления позволяют добиться порядка. Авторы должны думать о содержании, о том, что они пишут, не беспокоясь о конечном визуальном облике. Автоматизация обработки документа позволяет сэкономить время работы и улучшить качество подготовки документа, что может способствовать наиболее эффективной работе производства.

КРАТНОЕ СОДЕРЖАНИЕ РАБОТЫ

Стоит отметить, что на сегодняшний день существует довольно большое разнообразие программ для работы с текстовыми документами, которые различаются своими возможностями и функциями, но наиболее популярным (по числу использования пользователями) остается Microsoft Word [1].

При наличии большого количества достоинств, MS Word имеет и ряд недостатков, в частности он является платным продуктом, достаточно дорогостоящим. Форматирование документов большого объема в целом не удобно, занимает много времени, а так же вызывает множество зацикленностей и «зависаний». У данного продукта низкий уровень безопасности, велика вероятность потери документов. А так же отсутствует возможность работы с формулами, пользователь вынужден пользоваться сторонними дополнениями.

OpenOffice.org Writer — еще один мощный текстовый процессор, также обеспечивающий режим форматирования WYSIWYG [4]. Однако, возникают и другие недостатки системы: долгое время загрузки программы, построение таблиц возможно только по определенному стандарту, неудобная навигация и построение документа. Кроме того, возникают частые ошибки при работе, что может повредить документ.

LaTeX – наиболее популярный набор макрорасширений (или макропакет) системы компьютерной вёрстки TeX, который облегчает набор сложных документов [6]. К недостаткам также можно отнести и то, что продукт не являет программой типа WISIWIG. Но главное неудобство составляет то, что программа предполагает изначальное задание параметров, а не форматирование уже набранного текста, и сложна в обучении.

Целью разработки является создание программного продукта для автоматизированного форматирования текстовых документов, способного превзойти по удобству использования и стабильности работы существующих аналогов.

Для реализации поставленной цели необходимо:

1 Разработать технические требования к системе.

2 Разработать интерфейс системы для пользователя, обладающей высокой эргономичностью.

3 Разработать систему хранения данных.

4 Разработать диаграмму классов для реализации программных модулей.

5 Реализовать Yii-модуль согласно техническому заданию, UML-диаграммы классов и схемы базы данных.

Система должна предоставлять веб-сервис по автоматическому приведению текстового документа к требованиям по оформлению. Функционировать должна в виде сайта.

Система должна иметь 2 вида пользователей – «Администратор» и «Пользователь».

Для работы с системой в качестве «Пользователя» необходимо пройти регистрацию. Для регистрации в системе пользователь должен указать: e-mail, пароль.

При первом входе в систему пользователю предлагается прочитать инструкцию по подготовке файлов к предоставлению их системе и соглашением о том, что он не имеет претензий к работе системы, если предоставляет некорректный файл.

При входе в систему пользователь должен иметь возможности:

– изменить информацию о своей учётной записи;

– загрузить файл или несколько файлов, которые нужно привести в соответствие с требованиями по оформлению;

– выбрать требования в соответствии с которыми будет осуществлено исправление документа (в том числе и из числа уже созданных пользователем требований к оформлению);

– создать собственные требования к оформлению документа, путём перехода на страницу создания требований;

– просмотреть текущее состояние уже загруженных пользователем файлов;

– просмотреть результат работы системы. при наличии вопросов от системы ответить на них.

После загрузки файлов и выбора требований, пользователь нажимает кнопку перехода к окну «Текущее состояние загруженных файлов». Это список загруженных файлов с указанием состояния. Состояния могут быть следующие: новый, обрабатывается, готов.

Для работы с системой в качестве «Администратора» необходимо произвести вход в систему, вести логин и пароль. Администратор может просматривать статистические данные по посещению, по заказам со статусом «обрабатывается» и «готов» на текущий момент, а также историю. Так же администратор создает общедоступные требования к оформлению документа.

При работе с текстом выделим следующие структурные единицы: страница, колонтитул, основной текст, нумерация, заголовок, содержание (оглавление), рисунки, таблицы, формулы, списки, кавычки, сноски, примечания. Для каждой структурной единицы необходимо реализовать настройки параметров, в соответствии с заданными требованиями.

Проанализировав все эргономические требования, разработан интерфейс, который удовлетворяет условиям, целям, задачам и функциям дипломного проекта.

Для того, чтобы начать работу с системой необходимо перейти на вебсайт используя веб-браузер. После ввода адреса в адресную строку, по которому расположено приложение, на экране отобразится приветственная страница, представленная на рисуноке 1.

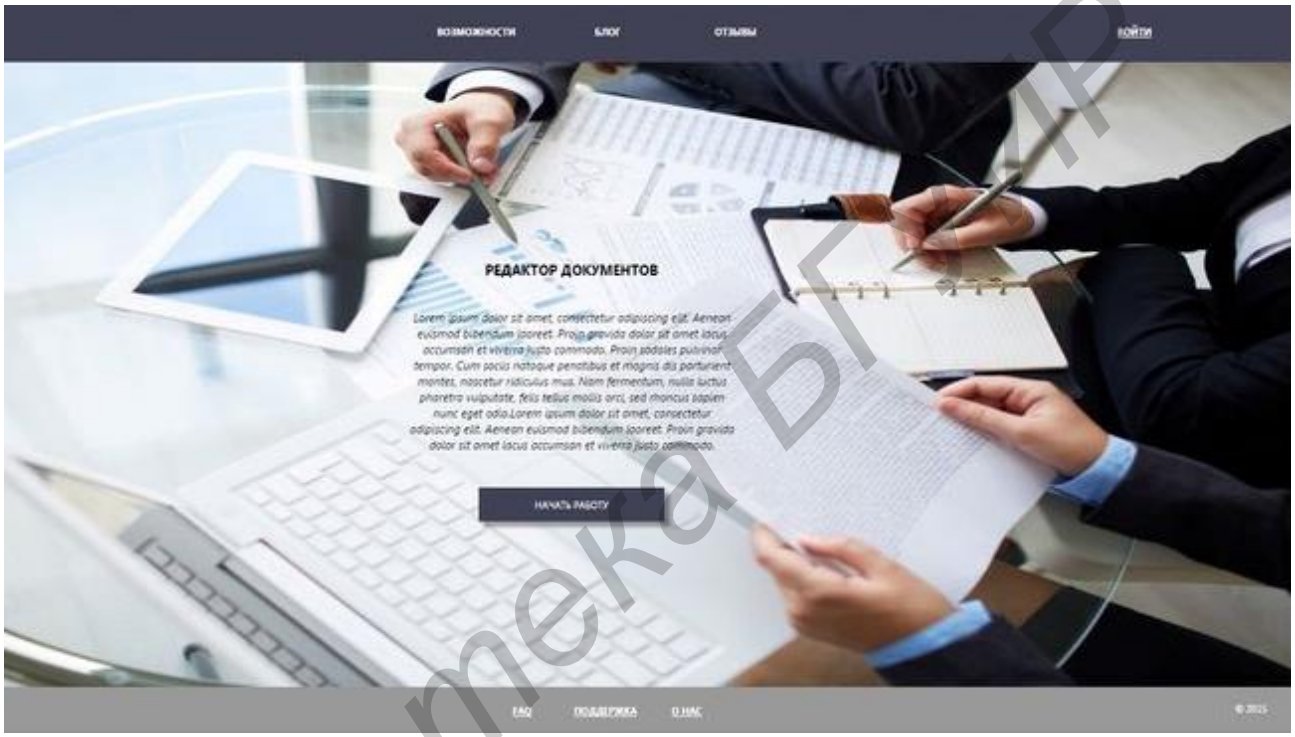

Рисунок 1 – Главная страница веб-сайта

В личном кабинете можно просмотреть сведения о действиях пользователя, в том числе список всех документов, которые он загружал. Список отображается в виде таблицы с основными характеристиками документа: «Название», «Размер», «Дата загрузки», «Состояние», «Требования».

Для каждого документа необходимы свои требования к оформлению. Они могут быть стандартные, например «Положение о диссертации на соискание степени магистра» для БГУИР, либо же собственные, которые может создать любой пользователь самостоятельно.

Таким образом, можно создать собственный набор шаблонных требований к текстовой документации, которые применяются только на данном производственном предприятии.

Процесс настройки требований представлен в виде блоков для различных частей текста с полями, выпадающими списками, таблицами выбора и так далее. Для каждого структурного элемента системы можно задать определенные параменты.

Интерфей интуитивно понятен и достаточно прост даже для пользователей впервые работающих с системой. После обработки документа пользователь получает страницу с данными о готовом файле, а также варианты дальнейшего действия: скачать, изменить, удалить или посмотреть результат обработки.

Для разработки программного модуля был выбран язык PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом, язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере [9]. Yii — это высокоэффективный, основанный на компонентной структуре PHP-фреймворк для быстрой разработки крупных веб-приложений [11]. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки.

PhpStorm – это интегрированная среда разработки на PHP с интеллектуальным редактором, которая «понимает» код, поддерживает PHP для современных и классических проектов, обеспечивает лучшее в индустрии автодополнение кода, рефакторинги, предотвращение ошибок налету и поддерживает смешивание языков [12].

MySQL – это система управления реляционными базами данных [13]. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных.

Прежде всего была спроектирована база данных. В данном случае она служит для сохранения настроек, при форматировании документов. В разделе 1.3 выделены основные технические требования предъявленные к системе. Исходя из специфики проекта, используемых инструментов и PHP-фреймворка, которые оговорены в предыдущем разделе, разработана и реализована в виде миграций база данных.

Для того, чтобы задавать тексту и отдельным его составляющим определенные параменты, необходимо четко их структурировать. Наиболее оптимальный способ – преобразовать имеющийся документ в html-документ. В таком виде текст представляет собой древовидную систему, в которой каждому элементу можно задать свои параменты и свойства.

OpenDocument Format, ODF – открытый формат файлов документов для хранения и обмена редактируемыми текстовыми документами (такими как заметки, отчёты и книги), электронными таблицами, рисунками, базами данных, презентациями. Является бесплатным продуктом, свободно распространяемый формат файла, представляет собой ZIP архив, содержащий несколько директорий и файлов, четыре из которых предназначены для содержимого документа, стилей, метаданных и настроек приложения. Данные хранятся в формате xml, что позволяет использовать для работы с ними соответствующие стандартные приемы.

В процессе разработки приложения было принято решение реализовать обработку ODT файлов по ряду причин:

– открытость документации;
– широкое распространие формата.

На рисунке 3.1 приведена типичная структура для файла ODT. По сути это Zip архив с вложеной структурой папок и xml файлов.

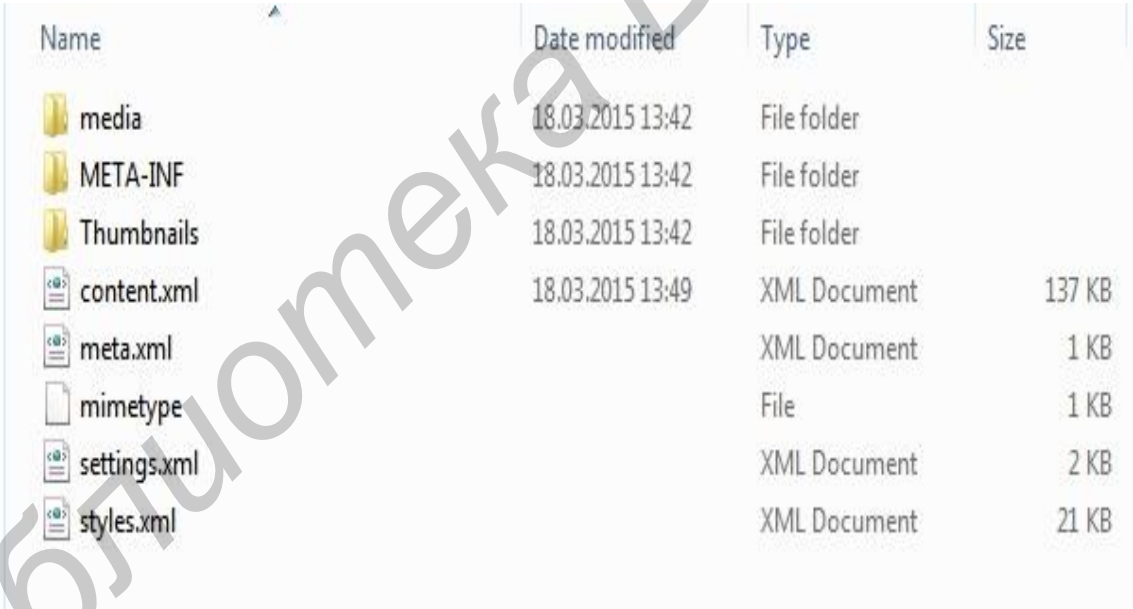

Рисунок 2 – Файл содержимого документа (content.xml)

Наибольший интерес для нас представляет файл content.xml в нем содержится структура документа и так называемые автостили.

Различные части текста в виде HTML-кода можно интрепретировать по-разному в зависимости от поставленных целей и заданных параметров.

Необходимо выделять заголовки по уровням. Заголовок первого уровня необходимо искать:

– согласно стилю настроек для вывода, и стилю в исходном документе;

– по правилам заложенным в инструкции по подготовке документа перед отправкой;

– по интеллектуальному анализу:

а) согласно наличию больших жирных букв;

б) согласно размеру предложения (до 10 слов);

в) согласно наличию цифр в начале предложения (цифры в начале предложения должны быть заменены на цифры, которые будут получены в результате анализа заголовков подзаголовков и установленного шаблона на нумерацию);

г) отсутствие текста, до и после строки;

Необходимо следить за наличием подписи к рисунку/таблице на той же странице, на которой рисунок находится. Выделять свободное место на странице, образовавшееся в результате добавления большого рисунка (автоматически заполнять это место, текстом со следующей страницы), автоматически изменять размеры рисунка (в случае, когда подпись к нему перескочила на следующую страницу), уведомляя пользователя (с указанием на сколько рисунок уменьшился). Автоматически переносить таблицу, если на странице осталась только шапка, с предложением довести текст на образовавшееся свободное место (или автоматически переставлять текст со следующей за таблицей страницы).

Для устранения свободного места предлагать пользователю (в виде настроек перед стартом анализа и редактированя):

– переносить абзац, следующий за рисунком на свободное место;

– предлагать дописать текст в свободное место;

– предлагать изменить размер рисунка, либо изменять размер автоматически, уведомляя пользователя; если картинка залазит на одну-полторы строки текста и рисунок занимает более 40% страницы, то тогда разрешить автоматическое уменьшение размера рисунка.

Для правильной постановке нумераций рисунков и таблиц необходимо отслеживать количество подписей к рисунку в тексте, реализовать автоматическое изменение номеров в тексте (ссылка на подпись к рисунку), при изменении кол-ва картинок (рисунков).

Так же необходимо удалять пустые пробельные строки.

Важно отслеживать лишние символы. Удаление символов лишнего переноса строк, когда текст форматируется не корректно. Необходимо склеивать строку в которой удален данный символ со следующей строкой, в случае, когда на следующей строке маленькая буква. В остальных случаях заменяем данный символ на символ «ввод». Так же удаление множественных пробелов, удаление множественных дефисов.

Особое внимание необходимо уделить таким элементам, как дефис, тире или длинное тире. Согласно настроек, заменяем дефис, тире и длинное тире, на символ, который мы приняли за тире.

Для простановки ссылок на список литературы в тексте, необходимо отыскать указания на источник, фигурные скобки с цифрой. Наличие точки относительно фигурных скобок, если точка находится до фигурной скобки, необходимо переставить её после указания номера литературы (после закрытой квадратной скобки).

Запись вида «[3,4,5,6,7,18,34]» приводится к записи вида «[3–7, 18, 34]» (возможно заменять перечисления на интервалы, если требуется, делать пробелы после запятых и прочее).

Часто в тексте необходимо указать единицы измерения. Алгоритм действий при их определении следующий: текст, написанный слитно с цифрой в конце, нужно сверить с таблицей единиц измерения, если строки совпадают, то применить к цифре настройку перевода её в разряд надстрочных символов

Текст, написанный слитно с цифрой в конце, сверить с таблицей единиц измерения. Если строки совпадают, то применить к цифре настройку перевода её в разряд надстрочных символов.

ЗАКЛЮЧЕНИЕ

В ходе написания магистерской диссертиции были составлены технические требования к системе, разработаны структурная схема программы и структура базы данных, разработаны алгоритмы работы пользователя. Разработан и реализован интерфейс системы, в соответствии с эргономическими требованиями, представляющий собой веб-сервис. Реализовано распознавание текста, автоматизированное разбиение его на структурные единицы и обозначение тегами.

Система разработана на языке PHP с использованием фреймворка Yii и базы данных MySQL.

Реализована библиотека по созданию индивидуальных шаблонов пользователями и форматирования производственной текстовой документации в соответствии с этими шаблонами.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

[1-А] Рубанова И.А. Управление распределенными системами / А.Ю. Пивоваров, И.А. Рубанова // 48-я научная конференция аспирантов, магистрантов и студентов БГУИР, Сб. докладов. – Мн.:БГУИР, – 2012 – С. 160.

[2-А] Рубанова И.А. Автоматизированная форматирование текстовой документации, И.А. Рубанова, В.С. Осипович // 51-я научная конференция аспирантов, магистрантов и студентов БГУИР, Сб. докладов. – Мн.:БГУИР, – 2015 – С. 71.

[3-А] Рубанова И.А. Форматирование текстовой документации, И.А. Рубанова, В.С. Осипович // 52-я научная конференция аспирантов, магистрантов и студентов БГУИР, Сб. докладов. – Мн.:БГУИР, – 2015.

[4-А] Рубанова И.А. Системы менеджмента и качества при форматировании производственной текстовой документации / И.А. Рубанова, В.С. Осипович // Научно-техническая конференция аспирантов, студентов и молодых специалистов «Новые информационные технологии в телекоммуникациях и почтовой связи», Сб. докладов. – Мн.:БГУИР, – 2015 – С. 269.