



Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Елонов Павел Сергеевич

Создание программной среды для создания расширяемых эссе

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
программист
А.А. Панфёров

Содержание

Введение	3
О расширяемом эссе	4
1 Постановка задачи	6
2 Проектирование программы для создания расширяемых эссе	7
2.1 Существующая реализация расширяемого эссе	7
2.2 Формат XML-документа для расширяемого эссе	7
2.3 Набор формул в существующей реализации	8
2.4 Общая схема работы программы	9
2.5 Разработка пользовательского интерфейса	9
3 Реализация системы для разработки расширяемых эссе	11
3.1 Выбор инструментальных средств	11
3.2 Внутреннее представление данных в программе для создания расширяемых эссе	11
3.3 Функциональность программы	12
3.4 Трансляция TeX-формул	12
3.5 Архитектура системы	13
3.6 Реализация программы для разработки расширяемых эссе	13
Заключение	15
Список литературы	16

Введение

На сегодняшний день Интернет получил широкое распространение, а самый распространенный вид информации в сети – это текст. Тексты разной направленности как правило имеют разную организацию материала. Это позволяет читателю быстро найти нужную и интересную ему информацию. Перечислим некоторые виды таких текстов.

- Простой текст. Не отличается от печатных изданий, обеспечивает читателю удобное чтение при качественном форматировании.
- Электронные энциклопедии. Данный вид организации позволяет читателю перемещаться на нужный раздел текущей статьи, а также дает возможность в любой момент перейти на другую статью, упоминающуюся в тексте.
- Электронные книги. С помощью оглавления они позволяют перейти на нужную главу. Читатель может создать в них закладку, чтобы при следующем открытии сразу мог перейти на нужное место в книге.
- Новостные ленты. Для каждой статьи высвечивается только их заголовок и несколько первых слов, а сами статьи разделены на рубрики. Это позволяет читателю быстро найти и прочесть интересные для себя новости.

В статье [1] предложен новый способ электронного представления текста – расширяемое эссе. Информация в расширяемом эссе представляется в некотором сжатом виде с возможностью уточнения и пояснения некоторых фрагментов при необходимости. В сфере образования данный способ изложения материала весьма востребован.

В настоящее время, в связи с развитием коммуникационных технологий, все более привлекательной выглядит концепция дистанционного образования. Системы дистанционного обучения позволяют преподавателям выкладывать материал по своему курсу в интернете, и все слушатели курса могут в любое время ознакомиться с ним. Системы дистанционного обучения предоставляют и другие возможности, например, преподаватели могут отслеживать успеваемость студентов с помощью них, назначая тесты и различные творческие задания. Каждый преподаватель сам выбирает форму, в которой он предоставляет материал: конспекты лекций в текстовом формате, видеоуроки и т.д. (см. [2]) У каждой формы представления материалов есть свои плюсы и минусы.

Но у всех видов представления материала есть одна общая черта – материал по всему курсу, как правило, довольно объемный. При подготовке к экзамену студент перечитывает весь материал по курсу полностью, а это занимает много времени. Более эффективным способом подготовки было бы просмотреть краткий конспект материала и более тщательно проработать непонятные фрагменты. Именно эту проблему может решить расширяемое эссе, так как читатель может сам выбрать, насколько подробное изложение материала ему необходимо.

О расширяемом эссе

Расширяемое эссе – краткое (в форме эссе) изложение материала, где для некоторых фрагментов возможно получить более подробную информацию, то есть расширение данного фрагмента. Расширения могут содержать в себе:

- более подробное изложение материала;
- пример, иллюстрирующий материал;
- библиографическую ссылку;
- контрольные вопросы по прочитанному.

Структура эссе следующая: весь текст делится на части (главы), каждая часть состоит из нескольких абзацев. Расширение относится к одному абзацу. В один момент времени отображается или базовый текст абзаца, или текст расширения, а также окружающий данный абзац текст. Каждая глава может содержать задачи к ней.

Реализация расширяемого эссе, представленная в статье [1], предлагает для представления текста расширяемого эссе использовать HTML-документ, отображаемый в удобочитаемом виде браузерами, а расширения в нем реализуются с помощью JavaScript.

Рассмотрим в качестве примера [3].



Затраты рандомизированного алгоритма, вообще говоря, не определяются однозначно входом алгоритма, они зависят и от полученных случайных чисел. Но *усредненные* затраты для каждого конкретного входа дают числовую функцию на множестве входов, что позволяет рассматривать сложность алгоритма, следуя общему определению сложности.

Рисунок 1: Пример абзаца со скрытым расширением



Затраты рандомизированного алгоритма при фиксированном входе, вообще говоря, не определяются однозначно, но зависят от полученных случайных чисел. В совокупности полученные случайные числа задают некоторый *сценарий* вычисления. При фиксированном входе мы можем рассмотреть множество всех сценариев и, приписав адекватным образом каждому из сценариев некоторую вероятность, ввести на полученном вероятностном пространстве случайную величину, значение которой для данного сценария равно соответствующим вычислительным затратам. Значение функции затрат на данном входе можно положить равным математическому ожиданию этой случайной величины (*усредненным* затратам для данного входа). После того как определена функция затрат и принято соглашение о том, что такое размер входа, мы можем, как обычно, рассматривать, например, сложность алгоритма в худшем случае.

Правомерен и такой подход к рандомизированным алгоритмам, при котором каждому возможному входу сопоставляется вероятностное пространство обычных детерминированных алгоритмов. Но каждый сценарий — это в определенном смысле и есть детерминированный алгоритм. Различие между этими подходами — терминологическое.

Рисунок 2: Пример абзаца с раскрытым расширением

Возьмем абзац, в котором говорится о сложности рандомизированных алгоритмов, изображенный на рисунке 1. На полях находится знак '+'. Это говорит о том, что можно нажать на этот значок и получить более развернутый текст.

После нажатия на '+' абзац будет выглядеть как на рисунке 2. Текст абзаца стал более развернутым, а значок '+' поменялся на стрелочку. Это означает, что абзац в данный момент раскрыт. На полях у нового абзаца мы видим знак ':', который говорит о том, что можно посмотреть пример.

Существующее решение для создания расширяемого эссе, предложенное в [1], заключается в создании XML-файла определенного формата. Этот процесс имеет несколько особенностей, усложняющих создание эссе:

- формат XML содержит в большом количестве теги (элементы языка разметки). Они не относятся непосредственно к тексту эссе, поэтому требуют от автора следить за структурой и корректностью XML-документа;
- вставка математических формул в текст эссе подразумевает приготовление изображений в виде файлов формата, поддерживаемого браузером (как правило, png или jpeg), и дальнейшую их вставку с помощью средств, предоставленных в шаблоне XSLT (набор специального тега для вставки изображения).

В силу этих обстоятельств, появилась потребность в программе с удобным интерфейсом, которая позволит создавать расширяемые эссе, используя существующую схему XML. Главная ее задача – скрыть от автора эссе все тонкости формирования XML и позволить ему целиком сосредоточиться на содержании эссе.

Часть 1

Постановка задачи

В рамках выпускной квалификационной работы необходимо создать программную среду для разработки расширяемых эссе. Для этого требуется:

1. Изучить существующее решение по созданию расширяемого эссе.
2. Спроектировать удобный пользовательский интерфейс разрабатываемой системы для создания расширяемых эссе.
3. Разработать внутреннее представление расширяемого эссе, удобное для работы с существующей реализацией.
4. Разработать механизм работы с математическими формулами.
5. Изучить существующие библиотеки для создания графического пользовательского интерфейса.
6. Создать программную реализацию системы.

Часть 2

Проектирование программы для создания расширяемых эссе

2.1 Существующая реализация расширяемого эссе

В статье [1] предложена реализация расширяемого эссе в виде HTML-документа, где для отображения расширений используются специальные кнопки, расположенные на полях. Отображение расширений реализовано с помощью JavaScript, который усложняет набор кода HTML-страницы с эссе. Поэтому было решено сначала создавать файл с разметкой эссе, а потом преобразовывать этот файл в HTML-страницу.

В основе данной реализации лежит технология XML, позволяющая создавать структурированные данные в текстовом формате. Изначально эссе представляет собой XML-файл специальной структуры, который потом по некоторым правилам превращается в HTML-документ.

Языки XML и HTML являются языками разметки и весьма похожи внешне (см. [4]), но их главное отличие заключается в том, что теги в XML несут информацию о своем содержимом, чем именно они являются. Это используется для машинного разбора документов.

Например, необходимо сохранить в документе список людей (их фамилию и имя). Если текст не будет структурирован, то возникнет сложность при его разборе: имя и фамилия записаны буквами, причем начиная с заглавной, и автоматизировано отличать их весьма затруднительно. XML-документ решит данную проблему: внутри тега Surname содержится фамилия человека, а внутри Name его имя. Теги языка HTML несут информацию лишь о том, как отобразить содержимое тега при просмотре документа в браузере.

Для преобразования документов XML в HTML существует специальный язык XSLT. Шаблон на этом языке задает набор правил, по которым определенные элементы в XML-документе заключаются внутри указанных HTML-тегов для дальнейшего отображения в браузере. Авторами статьи [1] подготовлен такой шаблон для отображения XML-файла с эссе в браузере.

2.2 Формат XML-документа для расширяемого эссе

Структура XML, предложенная в статье [1], предполагает в качестве корня документа использовать элемент `<Essay>`. Внутри этого тега находятся элементы:

- `<Title>` - может отсутствовать, определяет заголовок эссе;
- `<Parts>` - включает в себя один или несколько элементов `<Part>`, о структуре которого будет сказано ниже.

Теги `<Part>` разбивают эссе на последовательные части, внутри элемента `<Part>` присутствуют теги:

- `<Number>` - может отсутствовать, определяет номер данной части;
- `<Paragraph>` - один или несколько, соответствует одному абзацу эссе;
- `<Tasks>` - может отсутствовать, определяет вопросы к данной части.

Внутри каждого элемента `<Paragraph>` находятся следующие теги:

- `<Basic>` - определяет содержимое основного текста абзаца;
- `<Extension>` - может отсутствовать, определяет текст и тип расширения.

Элемент `<Extension>` имеет атрибут `<type>`. С помощью данного атрибута определяется тип расширения: «More», «Cite» и «Example» для расширений с развернутой информацией, библиографической ссылкой или примером соответственно. Может содержать один или несколько тегов `<Paragraph>`.

Элемент `<Tasks>` содержит один или несколько тегов `<Task>`. В свою очередь, внутри `<Task>` находятся элементы:

- `<Question>` - определяет текст вопроса;
- `<Options>` - определяет ответы на вопрос;
- `<Answer>` - порядковый номер правильного варианта ответа в элементе `<Options>`;
- `<Comment>` - пояснение к правильному ответу.

Элемент `<Options>` включает в себя несколько тегов `<Option>` с вариантами ответов на вопрос. Для форматирования текстов внутри абзацев предусмотрен ряд дополнительных тегов:

- `` - указывает, что текст должен быть выделен курсивом;
- `<bf>` - указывает, что текст должен быть выделен жирным шрифтом;
- `<p>` - указывает, что текст должен начинаться с новой строки;
- `<center>` - указывает, что текст должен располагаться по центру;
- `<indent>` - указывает, что перед текстом должен быть сделан отступ (несколько таких тегов подряд задают двойной, тройной и т.д. отступ);
- `` - позволяет вставить в текст изображение, расположенное в файле по заданному в атрибуте `src` адресу.

2.3 Набор формул в существующей реализации

Тексты на математическую тематику включают в себя много формул, и создание изображений с ними занимает довольно много времени. Например, для создания расширяемого эссе [3] потребовалось создать 751 изображение.

Основная целевая группа пользователей (составителей эссе) в большей своей части знакомы с системой компьютерной верстки \TeX , ее возможности по набору формул остаются непревзойденными. В связи с этим было принято решение использовать для набора формул в разработанной программе формат, использующийся в системе \TeX . \TeX широко используется математиками для создания различных публикаций, поэтому такой формат для набора формул будет привычным и удобным для многих авторов, желающих написать эссе математической направленности (см. [5], [6]).

2.4 Общая схема работы программы

Результат работы программы – это XML-файл с текстом эссе, разметка которого соответствует описанной в статье [1]. Далее этот файл может быть отображен в браузере с использованием шаблона XSLT, подготовленного авторами [1].

При работе программы эссе, над которым идет работа, представляется в оперативной памяти в виде специальных структур, которые будут описаны ниже. При наборе текста эссе автор может задавать структуру эссе, создавая главы и расширения для абзацев с помощью нажатия на соответствующие кнопки, при этом программа генерирует новые объекты внутреннего представления.

При сохранении эссе в файл программа переводит внутреннее представление эссе в XML-файл со структурой, описанной в статье [1], вставляя необходимые теги. При открытии этого файла браузером XSLT-шаблон переведет его в HTML-документ, и он отобразится в браузере.

2.5 Разработка пользовательского интерфейса

При разработке интерфейса необходимо учитывать два важных аспекта: нужды пользователя и сложившиеся стандарты оформления интерфейса в текстовых редакторах. Традиционно большую часть окна занимает поле текстового редактора, так как основная работа по набору и редактированию текста будет производиться именно в нем. Слева будет располагаться панель навигации, так как такое расположение является стандартным для большинства текстовых редакторов (см. [7]).

В панели навигации будут представлены главы и абзацы в виде дерева: на верхнем уровне – главы, у каждой главы свои абзацы, у абзацев могут быть свои расширения. Этот элемент управления будет изменяться автоматически при внесении изменений в текст эссе. Одна из важных ролей, которую будет выполнять панель навигации – это работа с расширениями. Для добавления расширения пользователю будет необходимо выбрать на панели абзац, который он хочет расширить, и нажать на соответствующую кнопку в главном окне программы. Так как пользователь будет сначала выбирать абзац, а потом нажимать на нужную кнопку, то логично разместить кнопки управления расширениями непосредственно под панелью навигации. Также с помощью данной панели будет производиться раскрытие расширений: если абзац имеет расширение, то при щелчке по соответствующему ему элементу навигации будет отображаться текст расширения. При показе текста расширений следует отображать тексты окружающих его абзацев, так как при раскрытии расширения текст эссе должен оставаться связным. На рисунке изображен примерный вид главного окна программы.



Рисунок 2.1: Макет окна приложения

Самый многочисленный элемент расширяемого эссе – абзац, поэтому создание абзацев происходит довольно часто. Пользователю будет неудобно каждый раз производить некоторое дополнительное действие для создания нового абзаца. Исходя из этого, процесс создания абзацев будет производиться без дополнительных действий со стороны пользователя – при наборе эссе текст будет делиться на абзацы автоматически.

Часть 3

Реализация системы для разработки расширяемых эссе

3.1 Выбор инструментальных средств

При выборе инструментов для реализации программы важны следующие аспекты: широкий набор элементов визуализации графического интерфейса и наличие классов для работы с XML.

Под данный набор параметров лучше всего подошла библиотека Qt. В ней представлено много элементов визуализации интерфейса и удобная схема взаимодействия между ними. Также эта библиотека имеет широкий набор классов для работы с различными данными, в том числе с XML (см. [8], [9]).

Выбор библиотеки Qt продиктовал выбор языка – C++. Для внутреннего представления расширяемого эссе удобно использовать объектно-ориентированную парадигму программирования. Также автор хорошо знаком с C++ и имеет опыт разработки на нем.

3.2 Внутреннее представление данных в программе для создания расширяемых эссе

В реализации расширяемого эссе, предложенной в статье [1], эссе представляется в виде XML-документа. Структура XML древовидна, поэтому логично хранить данные также в виде дерева, чтобы каждый элемент представлял собой специализированный класс (рисунок 3.1). При трансляции внутреннего представления в XML содержимое каждого элемента будет заключено в соответствующие теги.

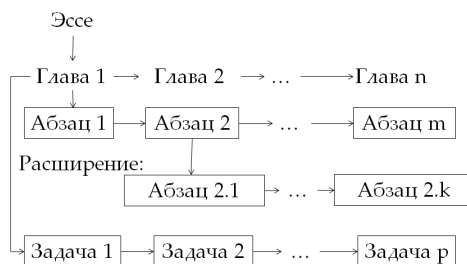


Рисунок 3.1: Внутреннее представление

3.3 Функциональность программы

Программа позволяет набирать текст эссе, при этом набранный текст сразу представляется в виде специальных структур. Добавление расширений, частей и контрольных вопросов производится по нажатию соответствующих кнопок на форме. В разработанной программе предусмотрена возможность загрузки и сохранения файлов. Загрузка производится из XML-файла со структурой расширяемого эссе, то есть можно загрузить уже созданное эссе и отредактировать его с помощью программы. Файлы сохраняются в XML с такой же структурой, потом этот файл можно преобразовать в HTML с помощью шаблона XSLT.

3.4 Трансляция \TeX -формул

Поддержка отображения формул в формате HTML появилась только в стандарте HTML5: в этот стандарт включен язык разметки MathML. Этот язык позволяет отображать формулы стандартными средствами браузера, но этот язык очень неудобен для набора формул вручную. В связи с этим, формулы набираются в формате \TeX , а затем преобразовываются для их дальнейшего отображения в HTML-файле. Для реализации трансляции формул в ходе работы были рассмотрены следующие варианты:

- автоматизировать трансляцию формул в изображения и затем вставлять специальным тегом в результирующий HTML;
- автоматизировать трансляцию формул в язык MathML и затем вставлять в результирующий HTML;
- использовать библиотеку MathJaxJS, автоматически распознающую формулы в HTML-файле, и затем отображающую формулы в браузере.

Первый и второй способы подразумевают наличие предустановленного транслятора \TeX на компьютере автора или использование облачных сервисов. Третий вариант требует лишь наличие библиотеки на компьютере или доступ к данной библиотеке через Интернет.

В разработанной программе используется вариант с MathJaxJS, так как он позволяет транслировать формулы без дополнительных действий.

3.5 Архитектура системы

Пользователь взаимодействует с программой через пользовательский интерфейс. Все изменения эссе, которые вносятся в процессе работы, сразу записываются во внутреннее представление. Для сохранения и загрузки файлов в формате XML используется специальный модуль трансляции из внутреннего представления в XML.

Результирующий XML-файл открывается браузером, который обращается к XSLT-шаблону и библиотеке MathJaxJS для формирования HTML-документа, представляющего собой расширяемое эссе. Затем этот файл отображается браузером и читатель может использовать все возможности расширяемого эссе.

На рисунке 3.2 изображено взаимодействие модулей системы между собой.

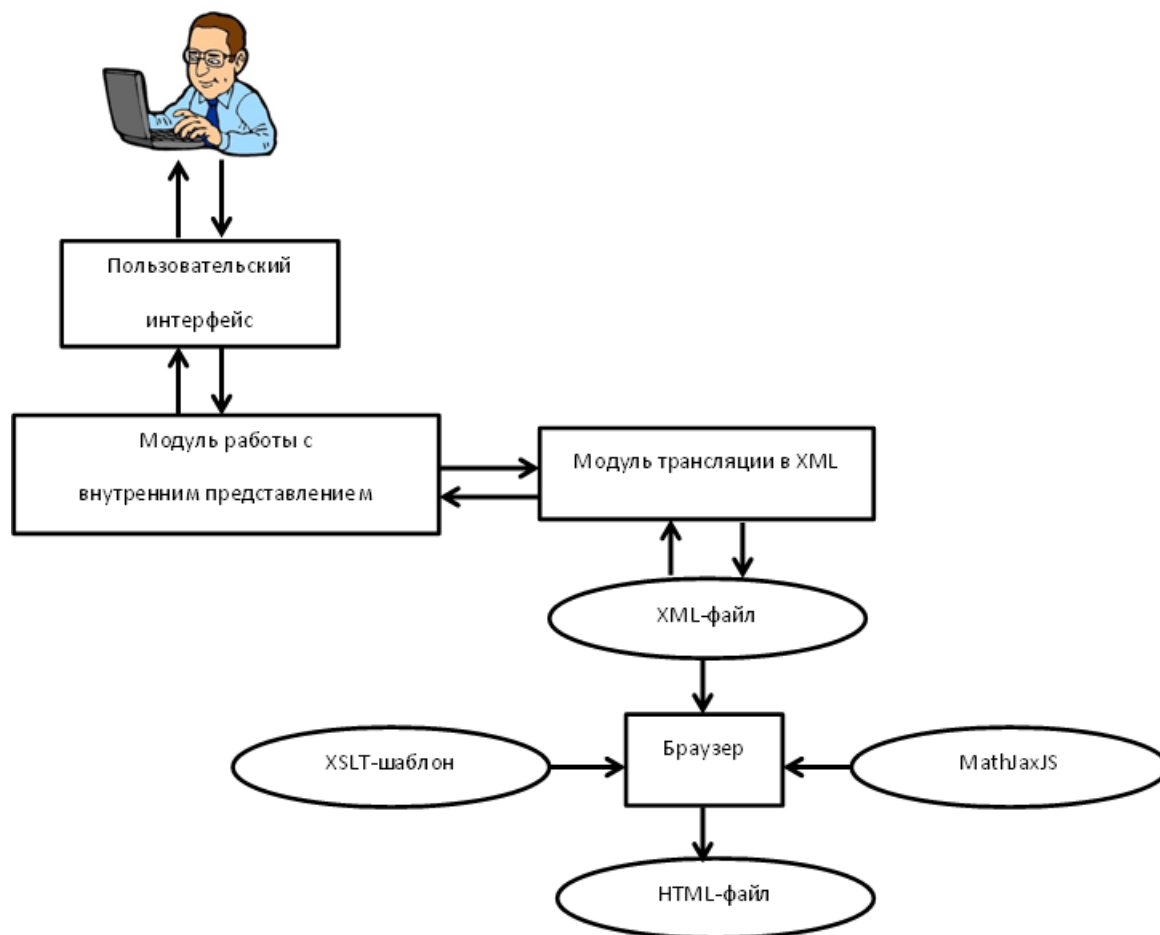


Рисунок 3.2: Архитектура системы

3.6 Реализация программы для разработки расширяемых эссе

В ходе работы была написана программа, которая позволяет пользователю создавать расширяемые эссе. Главная задача данной программы – позволить автору набирать текст расширяемого эссе, не задумываясь о служебных метках, не относящихся к содержательной части эссе. Интерфейс программы был разработан специально для решения данной задачи, автор эссе должен лишь указывать структуру эссе с помощью графического пользовательского интерфейса.

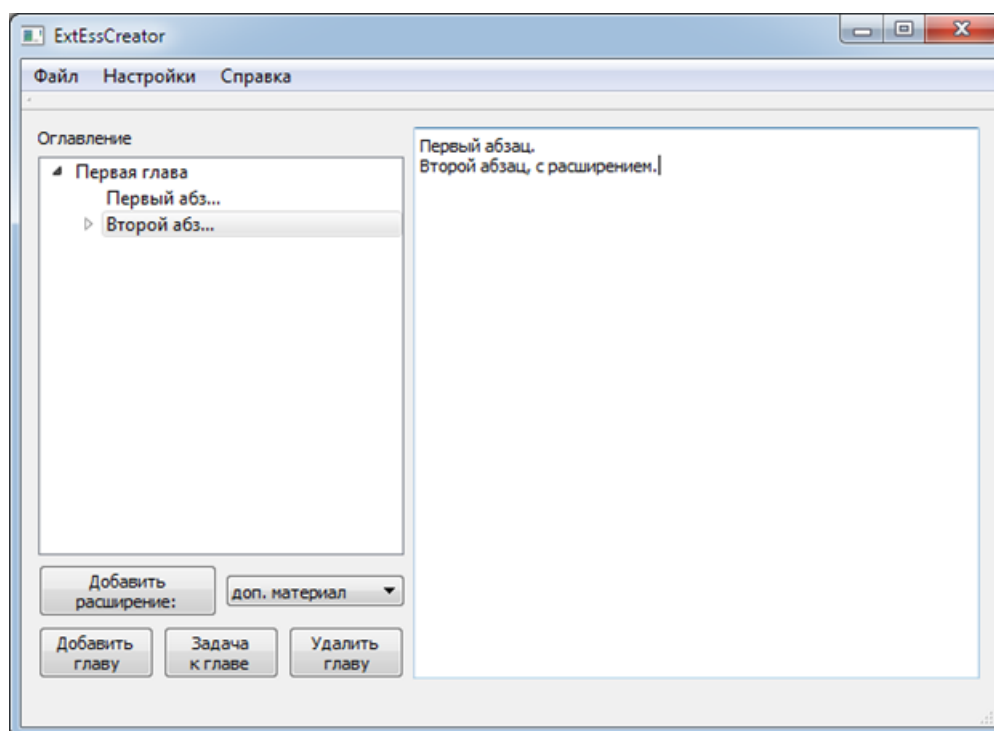


Рисунок 3.3: Главное окно

На рисунке изображено главное окно программы. В меню «Файл» находятся пункты для сохранения и загрузки расширяемого эссе, в «Настройках» указывается путь до XSLT-шаблона и настройки работы для библиотеки MathJaxJS. В меню «Справка» доступно краткое руководство для пользователя.

Слева находится панель навигации. В оглавлении на верхнем уровне указана глава, а далее абзацы, содержащиеся в ней. Около второго абзаца есть значок, который позволяет раскрыть этот элемент. Это означает, что у данного абзаца есть расширение. При щелчке по абзацу в панели навигации, в окне редактора курсор установится на начало этого абзаца и окно прокрутится так, чтобы курсор был виден. Данное решение призвано облегчить редактирование текста при создании объемных эссе, потому что поможет быстро находить абзацы, в которых потребовалась правка.

Справа находится поле текстового редактора, в нем отображается текст текущей главы. Для того, чтобы посмотреть текст расширения, необходимо раскрыть элемент в оглавлении. При этом базовый текст абзаца поменяется на расширенный и будет отображаться с отступом слева, чтобы визуально выделяться в тексте.

Под панелью навигации находятся кнопки, с помощью которых можно добавлять расширения нужного типа, а также добавлять и удалять главы эссе.

Заключение

В ходе выпускной квалификационной работы были получены следующие результаты:

1. Разработан пользовательский интерфейс для набора расширяемых эссе.
2. Спроектировано внутреннее представление для расширяемого эссе.
3. Предложена и реализована работа с формулами на основе библиотеки MathJaxJS в формате \TeX .
4. Реализована система для набора расширяемых эссе.

Возможно дальнейшее развитие данной системы, так как некоторые функции расширяемого эссе, такие как вставка изображений, возможность форматирования текста и указание связей между расширениями, остались нереализованными. Развитие системы так же возможно посредством добавления альтернативных способов трансляции формул в формате \TeX в графические изображения. Можно использовать возможности стандарта языка HTML5 для отображения формул в браузере, но для этого требуется корректировка XSLT-шаблона, предложенного в [1].

Список литературы

1. Абрамов, С. А. Расширяемое эссе как гипертекстовая схема информационного и учебного материала / С. А. Абрамов, Е. А. Бордаченкова, Д. Е. Хмельнов. // Журнал вычислительной математики и математической физики. – 2013. – №3. – С. 495-501.
2. Гильмутдинов, А. Х. Электронное образование на платформе Moodle / А. Х. Гильмутдинов, Р. А. Ибрагимов, И. В. Цивильский. – Казань: Казанский Государственный Университет, 2008. – 169 с.
3. Сложность алгоритмов [Электронный ресурс]. – Режим доступа: <http://www.ccas.ru/ca/essay>, свободный. – Загл. с экрана.
4. Холзнер, С. XSLT. Библиотека программиста / С. Холзнер. – СПб.: Питер, 2002. – 544 с.
5. Львовский, С. М. Набор и верстка в системе LaTeX / С. М. Львовский. – М.: Московский центр непрерывного математического образования, 2003. – 448 с.
6. Столяров, А. В. Сверстай диплом красиво: LaTeX за три дня / А. В. Столяров. – М.: МАКС Пресс, 2010. – 101 с.
7. Джонсон, Дж. Простые приемы разработки пользовательских интерфейсов / Дж. Джонсон. – СПб. : Питер, 2012. – 224 с.
8. Бланшет, Ж. Qt 4. Программирование GUI на C++ / Ж. Бланшет, М. Саммерфилд. – СПб.: КУДИЦ-Пресс, 2008. – 736 с.
9. Саммерфилд, М. Qt. Профессиональное программирование. Разработка кроссплатформенных приложений на C++ / М. Саммерфилд. – СПб.: Символ-Плюс, 2011. – 552 с.