

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение
высшего образования

«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Факультет программной инженерии и компьютерной техники

Направление (специальность) — 09.04.02 Информационные системы и технологии

Специализация — Веб-технологии

Дисциплина — Проектирование и анализ языков веб-решений

Курсовой проект (работа)

ТЕМА: Проектирование XSL-шаблона для визуального отображения
структуры математического информационного ресурса

ВЫПОЛНИЛ

Студент группы

P41622

№ группы

подпись, дата

Разина В.Д.

ФИО

ПРОВЕРИЛ

ученая степень, должность

подпись, дата

Государев И.Б.

ФИО

САНКТ-ПЕТЕРБУРГ

2019 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Глава 1. Язык программирования XSL.....	5
1.1 Понятие и особенности XSL.....	5
1.2 XPath.....	6
1.3 XSL Transformation.....	10
Глава 2. Основы MathML.....	15
Глава 3. Проектирование шаблона.....	18
ЗАКЛЮЧЕНИЕ.....	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	24

ВВЕДЕНИЕ

В наши дни большая часть информации поступает, хранится и изучается в электронном виде. Заметные изменения произошли и в сфере научного обмена, в результате современного исследователя невозможно представить без компьютера, который из средства набора текстов давно стал инструментом получения знаний. В связи с этим, проблема представления и обработки математических текстов в электронной форме приобрела особую актуальность. В существенной части эта проблема касается представления математических формул. Представление математических документов в pdf-формате или html-файлах, когда каждая формула является ссылкой на графический ресурс, делает структурную обработку такой информации крайне затруднительной. Поэтому наиболее распространенное на данный момент решение – представление формул в виде графических файлов – неудовлетворительно с точки зрения структурной обработки математических текстов.

Для математического сообщества наибольший интерес представляет MathML (Mathematical Markup Language) – технология, предназначенная для представления математических формул. Разработка этой технологии ведется консорциумом W3C с 1999 года. Поскольку технологии семантического веба предоставляют новый способ организации веб-информации, который дает возможность на более высоком уровне решать задачи программной обработки документов, MathML изменяет принципы организации и управления электронными публикациями по математике [10].

В свою очередь, универсальным механизмом управления отображением XML-документов является расширяемый язык таблиц стилей XSL (eXtensible Stylesheet Language). С помощью XSL можно трансформировать XML документ в любой формат, например HTML и, в частности, в XML. На языке XSL можно описать, как будет оформлен итоговый документ, где и как должны располагаться данные [2]. Используя языки XSL и MathML, можно

создать шаблон, который будет отображать структуру математической формулы или иного выражения.

В соответствии с вышеизложенным, цель настоящей работы состоит в проектировании XSL-шаблона для визуального отображения структуры математического информационного ресурса.

Для достижения поставленной цели необходимо решить ряд задач:

- 1) Изучить XSL: определить понятие и особенности данного языка; проанализировать XPath и XSLT;
- 2) рассмотреть основы MathML;
- 3) непосредственно разработать шаблон для отображения структуры математических выражений.

ГЛАВА 1. Язык программирования XSL

1.1 Понятие и особенности XSL

XSL (eXtensible Stylesheet Language) – расширяемый язык таблиц стилей – механизм управления отображением XML-документов, представляющий собой совокупность рекомендаций консорциума Всемирной паутины (W3C)[1]. XSL позволяет открывать XML-документ непосредственно в браузере без посредничества HTML-страницы; осуществлять отбор и сортировку данных XML при их отображении, предоставляет доступ ко всем компонентам XML (элементам, атрибутам, комментариям и инструкциям по обработке), даёт возможность включать в таблицу стилей сценарии.

XSL состоит из следующих частей:

- 1) XSL Transformations (XSLT) — язык преобразований XML-документов. В общих словах процесс преобразования можно описать следующим образом — XSLT преобразовывает исходное дерево XML в XML дерево-результат (см. Рисунок 1). На данный момент актуальна версия XSLT 3.0, одобренная 8 июня 2017 года.

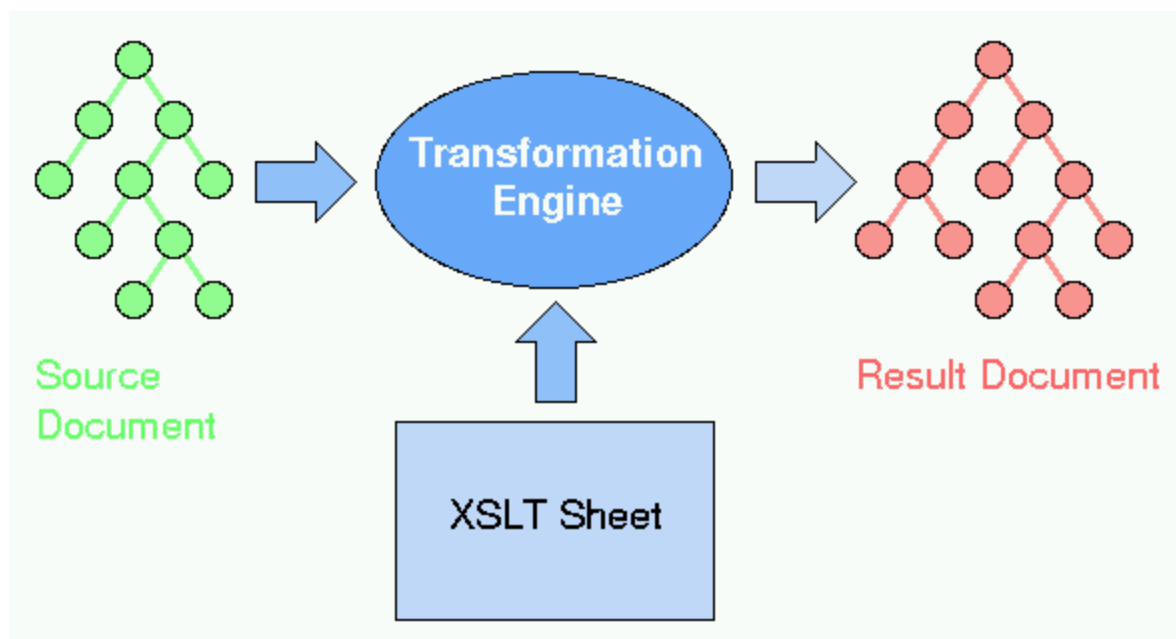


Рисунок.1 Процесс преобразования

Для навигации по документу XSLT использует XPath.

- 2) XPath — язык путей и выражений, используемый в XSLT для доступа к отдельным частям XML-документа. В процессе преобразования XSLT при помощи XPath определяет те части исходного документа, которые должны соответствовать одному или нескольким предопределенным шаблонам [3]. Если соответствие будет найдено, то XSLT преобразует эту часть исходного документа и создаст конечный документ.
- 3) XSL Formatting Objects (XSL-FO) – язык разметки типографских макетов и иных печатных материалов. Однако в 2012 году разработка XSL-FO была остановлена, и, представив в 2013 CSS3 (каскадные таблицы стилей – формальный язык описания внешнего вида документа, написанного с использованием языка разметки), консорциум W3C определил новый стандарт форматирования документов. Другими словами, с 2013 года CSS3 позиционируется как замена XSL-FO [4].

1.2 XPath

Перед проектированием XSL-шаблона, прежде всего, необходимо рассмотреть XPath. Не владея основами XPath, невозможно создавать и использовать XSLT- документы.

Рассмотрим основы XPath на следующем XML-документе:

```
<?xml version="1.0" encoding="UTF-8"?>
<cinema>
  <film>
    <title lang="en">The Thing</title>
    <director>John Carpenter</director>
    <year>1982</year>
    <genre>horror</genre>
```

</film>

</cinema>

В XPath существует семь видов узлов: элемент, атрибут, текст, пространство имён, инструкции обработки, комментарии и узлы документа [5]. В исследуемом примере можно выделить следующие виды:

<cinema>	корневой элемент
<director>John Carpenter </director>	узел
lang="en"	атрибут

Отношения узлов

Отношение	Описание	XML-документ
Родитель	film –родитель для элементов title, director, year, genre	<?xml version="1.0" encoding="UTF-8"?> <cinema> <film> <title lang="en">The Thing</title> <director>John Carpenter</director> <year>1982</year> <genre>horror</genre> </film> </cinema>
Потомки	title, director, year, genre – потомки элемента film	
Элементы одного уровня	Узлы, имеющие одного родителя (title, director, year, genre)	
Предки	Родитель узла, родитель родителя узла и т.д. Предки элемента title – film, cinema	

Добавим в пример еще один фильм и рассмотрим синтаксис XPath. XPath использует выражения пути для выбора узлов или множества узлов в документе XML. Узел можно выбрать, следуя пути или по шагам.

Выражение	Результат	XML-документ
cinema	Выбирает все узлы с этим именем	<pre><?xml version="1.0" encoding="UTF-8"?> <cinema> <film> <title lang="en">The Thing</title> <director>John Carpenter</director> <year>1982</year> <genre>horror</genre> </film> <film> <title lang="en">It</title> <director>Andres Muschietti</director> <year>2017</year> <genre>horror</genre> </film> </cinema></pre>
/cinema	Выбирает корневой элемент	
//film	Выбирает все элементы film, независимо от того, где они находятся	
cinema/film	Выбирает все фильмы, которые являются потомками элемента cinema	
/cinema/*	Выбирает все дочерние узлы элементы cinema	
//@lang	Выбирает атрибуты, которые называются lang	

Предикаты (используются для поиска специфического узла или узла, содержащего специфическое значение)		
/cinema/film[1]	Выбирает первый элемент фильм, который является потомком элемента cinema	
/cinema/film[year>2000]	Выбирает все фильмы в элементе cinema, которые имеют элемент year со значением больше 2000	
/cinema/film[last()]	Выбирает последний фильм, который является дочерним элементом cinema	

Можно не использовать названия узлов, используя следующие приемы:

.	Выбирает текущий узел
..	Выбирает родителей текущего узла
*	Любой узел

@*	Любой узел-атрибут
//*	Выбирает все элементы в документе

После анализа основных положений XPath, можно переходить к рассмотрению языка XSLT как языка.

1.3 XSLT

Язык преобразований XSLT (от англ. eXtensible Stylesheet Language Transformations) – дословно переводится как расширяемый язык преобразования таблиц стилей, но правильнее его называть языком для преобразования структуры документов. Являясь одним из реализаций языка XML, технология служит транслятором XML-документов, с помощью которого можно свободно модифицировать исходный текст, и трансформировать его в другие языки [6].

Фактическая сборка результирующего документа происходит, когда исходный документ и таблица стилей XSL передаются в синтаксический анализатор XSLT (XSLT-процессор). Этот процесс можно разбить на три составляющих:

1. Обращение к преобразуемому объекту;
2. Создание результата преобразования;
3. Логика, связывающая первые два действия и направляющая процесс преобразования [7].

Применительно к преобразованию XML-документов первая подзадача означает получение информации, содержащейся в документе – в том числе и информации о структуре, которая является неотъемлемой его частью. Обращение в данном случае имеет несколько смыслов, в том числе – опрашивать, делать запросы, вычислять, выбирать; в общем смысле – задавать о документе вопросы и получать на них ответы. Для этой цели в XSLT используется, как средство обращения, XPath – язык путей в XML-

документах. Он определяет те части исходного документа, которые соответствуют одному или более заранее заданным шаблонам. Когда такое соответствие обнаруживается, XSLT преобразует соответствующую часть документа, а те компоненты, которые не соответствуют – попадают в результат не модифицированными или же вовсе пропускаются. Остальные условные части преобразования являются прерогативой самого XSLT [8].

Следует отметить, что преобразование в XSLT не является последовательностью действий, которую необходимо выполнить для достижения результата. Преобразование — это набор шаблонных правил, каждое из которых определяет процедуру обработки определённой части документа. Иными словами, преобразование в XSLT объявляет, декларирует правила преобразования – шаблоны, которые применяются к входящему документу. Таким образом, структурной единицей таблицы стилей является шаблон [9]. XSLT-процессор в конечном итоге генерирует выходящий документ, являющийся целью преобразования (см. Рисунок 2).

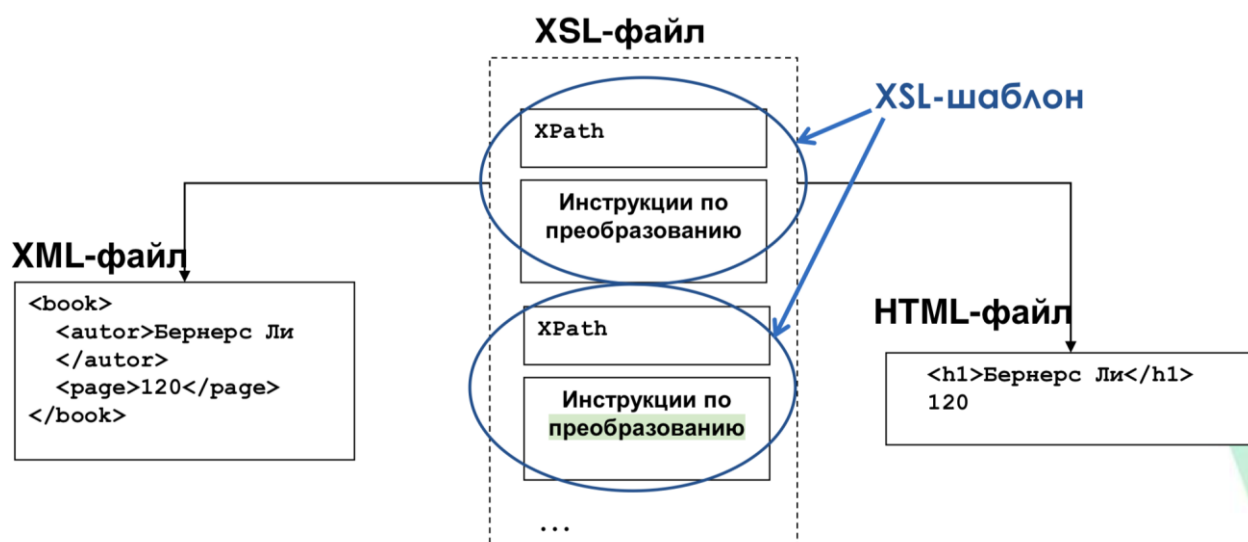


Рисунок 2. Процесс XSL Transformations

Существуют два основных шага для отображения XML-документа при использовании XSL-таблицы стилей:

Создание файла XSL-таблицы стилей. XSL является приложением XML, т.е. XSL-таблица представляет собой корректно сформированный XML-документ, который отвечает правилам XSL.

Связывание XSL-таблицы стилей с XML-документом. В XML-документ включается инструкция по обработке xml-stylesheet, которая имеет следующую форму записи:

```
<?xml-stylesheet type="text/xsl" href=xslFileURL?>
```

Предположим, что нам нужно преобразовать XML документ. Для этого добавляем в него ссылку на таблицу стилей XSL. Первые 2 строчки данного документа будут выглядеть следующим образом:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml-stylesheet type="text/xsl" href="//konfucij.ru/cd_catalog.xsl"?>
```

Где **//konfucij.ru/cd_catalog.xsl** – адрес XSL файла.

Перейдем к его содержимому. Так как таблица стилей XSL это XML документ, он всегда должен начинаться с XML декларации: **<?xml version="1.0" encoding="UTF-8"?>**. Чтобы получить доступ к элементам, атрибутам и другим функциям XSLT, необходимо в начале документа декларировать пространство имен XSLT.

Согласно рекомендации консорциума W3C таблица стилей XSL декларируется следующим образом:

```
<xsl:stylesheet version="1.0"
```

```
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

Строка **xmlns:xsl="http://www.w3.org/1999/XSL/Transform"** указывает на официальное пространство имен XSLT консорциума W3C. Если использовать это пространство имен, то также необходимо указывать и атрибут **version="1.0"**.

Таблица стилей XSL содержит один или больше наборов правил преобразования, которые называются шаблонами преобразования. Шаблон преобразования содержит правила, которые применяются, когда найден узел, соответствующий условию поиска.

Для создания шаблонов преобразования используется элемент **<xsl:template>**.

При этом атрибут **match** используется для ассоциации шаблона с XML элементом. Также, атрибут **match** может использоваться, чтобы определить шаблон для всего XML документа целиком. Значение атрибута **match** это выражение XPath (например, **match="/"** определяет весь документ).

Содержимое элемента **<xsl:template>** определяет некий HTML код, который записывается в выходной документ. Таким образом, каждый шаблон содержит:

адрес узлов (на XPath), на которые действует данный шаблон (записывается в атрибут **match** элемента **xsl:template**); инструкции по преобразованию

Пример:

```
<xsl:template match="путь">  
    инструкции по преобразованию  
</xsl:template>
```

Элемент **<xsl:value-of>** используется для извлечения значения отобранного XML элемента и добавления его в выходной поток преобразовываемого документа.

Атрибут **select** элемента **<xsl:value-of>** определяет, какое именно выражение будет использовано. Select содержит выражение XPath, которое работает, как навигация по файловой системе; прямая косая черта (/) выбирает поддиректории.

Элемент **<xsl:for-each>** позволяет организовывать циклы в процессе XSLT преобразования. XSL элемент **<xsl:for-each>** может использоваться для выбора каждого XML элемента заданного узлового набора.

Также, можно осуществлять фильтрацию вывода из XML файла. Для этого следует добавить нужный критерий к атрибуту **select** элемента **<xsl:for-each>**.

Пример:

```
<xsl:for-each select="cinema/film[director='John Carpenter']">
```

Допустимые операторы фильтрации:

- = (равно)
- != (не равно)
- < (меньше чем)
- > (больше чем)

Элемент **<xsl:apply-templates>** применяет некий шаблон к текущему элементу или к дочернему узлу текущего элемента.

Если в элемент **<xsl:apply-templates>** добавить атрибут **select**, то он будет относиться только к дочернему элементу, который соответствует значению этого атрибута. Таким образом, атрибут **select** может использоваться для определения порядка, в котором будут обрабатываться дочерние узлы.

Рассмотрев XSL и его составляющие, можно перейти к изучению MathML.

Глава 2. Основы MathML

Основная трудность при внедрении математики в World Wide Web состоит в том, чтобы зафиксировать как представление, так и содержание (то есть значение) не теряя высокоразвитую знаковую систему математики и возможность взаимосвязи. Консорциум World Wide Web (W3C) – организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины – в 1994 году создает рабочую группу по разработке математической разметки. Была предложена спецификация MathML, позволяющая представлять математический материал в двух видах: презентационная и содержательная разметка [10].

пример	презентационный	содержательный
a-b	$\langle \text{mrow} \rangle \langle \text{mi} \rangle a \langle \text{mi} \rangle$ $\langle \text{mo} \rangle - \langle \text{mo} \rangle \langle \text{mi} \rangle b \langle \text{mi} \rangle$ $\langle \text{mrow} \rangle$	$\langle \text{apply} \rangle \langle \text{minus} \rangle \langle \text{ci} \rangle a \langle \text{ci} \rangle \langle \text{ci} \rangle b \langle \text{ci} \rangle$ $\langle \text{apply} \rangle$

MathML предлагает гибкую и расширяемую систему записи математического материала, позволяющей взаимодействовать с внешними программами и осуществлять высококачественное отображение в различных информационных средах. Для создания, конвертирования, просмотра, печати документов MathML предлагаются специальные редакторы формул, конвертеры и другие специализированные программные средства [11].

Элементом верхнего уровня в MathML является тэг `<math>`. Каждый допустимый экземпляр MathML должен быть внутри этого контейнера. Внутри может быть произвольное число других дочерних элементов [12].

Рассмотрим ряд основных элементов, которые необходимы для представления структуры математического выражения.

Элемент MathML(тег)	Описание	Пример	Результат
<code><mi></code>	Идентификатор (например, имена функций, переменные или символьные константы)	<code><mi>a</mi></code>	a
<code><mn></code>	Число. Допускается наличие в нем текста, который на самом деле является числовой величиной, например "семь".	<code><mn>1234</mn></code>	1234
<code><mo></code>	оператор	<code><mi>a</mi> <mo>+</mo> <mi>b</mi></code>	a+b
<code><msqrt></code>	Квадратный корень	<code><msqrt> <mi>a</mi> <mo>+</mo> <mi>b</mi> </msqrt></code>	$\sqrt{a + b}$
<code><mfrac></code>	Формирует дробь из 2 подэлементов	<code><mfrac> <mi> a </mi> <mi> b </mi> </mfrac></code>	$\frac{a}{b}$

<code><msub></code>	нижний индекс	<code><msub></code> <code><mi>x</mi></code> <code><mn>1</mn></code> <code></msub></code>	x_1
<code>msup</code>	Верхний индекс	<code><msup></code> <code><mi>x</mi></code> <code><mn>12</mn></code> <code></msup></code>	x^{12}
<code><munder></code>	Добавляет символ под основанием	<code><munder></code> <code><mi>X</mi></code> <code><mi>&#x23DE;</mi></code> <code></munder></code>	$\underset{\sim}{X}$
<code><mover></code>	Добавляет символ над основанием	<code><mover></code> <code><mi>X</mi></code> <code><mi>&#x23DE;</mi></code> <code></mover></code>	\tilde{X}
<code><munderover></code>	Добавляет символы одновременно и над, и под основанием	<code><munderover></code> <code><mo> &#x222B; </mo></code> <code><mn> 0 </mn></code> <code><mi> &#x221E; </mi></code> <code></munderover></code> <code><mi>x</mi></code>	$\int_0^{\infty} x$
<code><mrow></code>	Для группировки подвыражений, которые обычно содержат один или несколько операторов с соответствующими операндами (например, <code><mi></code> и <code><mn></code>). Этот элемент отображается как горизонтальная строка, содержащая его аргументы.	<code><mrow></code> <code><mn> 1 </mn></code> <code><mo> + </mo></code> <code><mn> 1 </mn></code> <code></mrow></code>	$1+1$

Глава 3. Проектирование шаблона

Пусть у нас есть следующее поддерево разметки, сделанной на выдуманном математическом языке, но близком к реальности:

```
<строка>
  <операнд>σ</операнд>
  <оператор>=</оператор>
    <корень>
      <строка>
        <дробь>
          <строка>
            <число>1</число>
          </строка>
          <строка>
            <операнд>N</операнд>
          </строка>
        </дробь>
      <строка>
        <низверх>
          <строка>
            <оператор>Σ</оператор>
          </строка>
          <строка>
            <операнд>i</операнд>
            <оператор>=</оператор>
            <число>1</число>
          </строка>
          <строка>
            <операнд>N</операнд>
          </строка>
        </низверх>
      <строка>
        <верх>
          <строка>
            <оператор>(</оператор>
          <низ>
            <строка>
              <операнд>x</операнд>
            </строка>
          <строка>
            <операнд>i</операнд>
          </строка>
        </верх>
      </строка>
    </корень>
  </строка>
</строка>
```

```

<оператор>—</оператор>
<операнд>μ</операнд>
<оператор >)</оператор>
</строка>
<строка>
    <число>2</число>
</строка>
</верх>
</строка>
</строка>
</строка>
</корень>
<оператор>.</оператор>
</строка>

```

Необходимо написать XSLT-стили, которые преобразуют это поддерево в поддерево элемента `math` с его последующей обработкой по технологии MathJax.

MathJax позволяет включать математику в веб-страницы, используя LaTeX, MathML или asciimath нотации, и математика будет обрабатываться с помощью JavaScript для создания HTML, SVG или MathML уравнений для просмотра в любом современном браузере.

Существует два способа доступа к MathJax: самый простой способ - использовать копию MathJax, доступную из сетевой службы, такой как `cdnjs.com`, но вы также можете загрузить и установить копию MathJax на свой собственный сервер или использовать ее локально на жестком диске (без необходимости доступа к сети) [13]. Был выбран первый способ доступа.

```

<script type="text/javascript" async
src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.5/MathJax.js?config=TeX-
MML-AM_CHTML">
</script>

```

Рекомендуется вставить данный скрипт в блок <head> документа. При необходимости он также может входить в <body>, но в рекомендации отмечают, что предпочтительнее <head>. Таким образом, будет загружена версия 2.7.5 MathJax с распределенного сервера и настроит его для распознавания математики в нотации не только MathML, но и AsciiMath, TeX [14].

Применив на практике полученные в результате анализа XSL и MathML знания, был составлен следующий XSL-документ:

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <head>
    <script type="text/javascript" async
src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.5/MathJax.js?config=TeX-
MML-AM_CHTML">
</script>
    <title>Курсовая работа</title>
  </head>
  <body>
    <h1 align="center">
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <xsl:apply-templates/>
</math>
    </h1>
  </body>
</html>
</xsl:template>

<xsl:template match="строка">
  <mrow> <xsl:apply-templates/> </mrow>
</xsl:template>

<xsl:template match="операнд">
  <mi> <xsl:value-of select="."/> </mi>
</xsl:template>
```

```

<xsl:template match="оператор">
    <mo> <xsl:value-of select="."/> </mo>
</xsl:template>

<xsl:template match="корень">
    <msqrt> <xsl:apply-templates/> </msqrt>
</xsl:template>

<xsl:template match="дробь">
    <mfrac> <xsl:apply-templates/> </mfrac>
</xsl:template>

<xsl:template match="число">
    <mn> <xsl:value-of select="."/> </mn>
</xsl:template>

<xsl:template match="низверх">
    <munderover> <xsl:apply-templates/> </munderover>
</xsl:template>

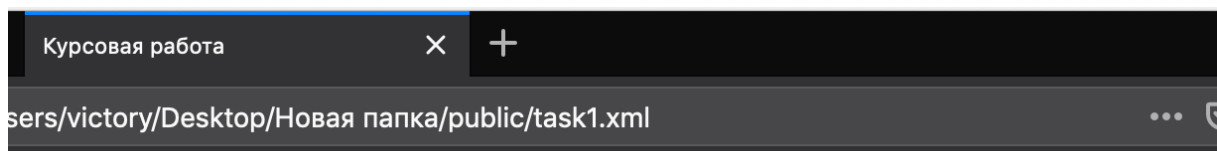
<xsl:template match="верх">
    <msup> <xsl:apply-templates/> </msup>
</xsl:template>

<xsl:template match="низ">
    <msub> <xsl:apply-templates/> </msub>
</xsl:template>

</xsl:stylesheet>

```

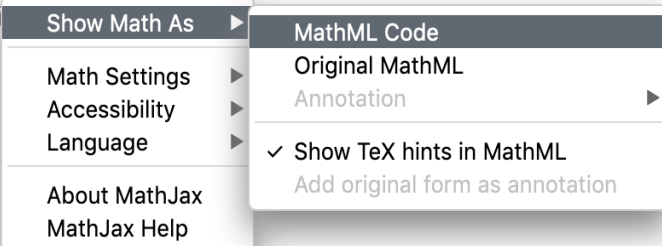
Таблица стилей была подключена к XML документу с рассматриваемым поддеревом. В результате в веб-браузере стало возможным увидеть математическую формулу (см.Рис.3)

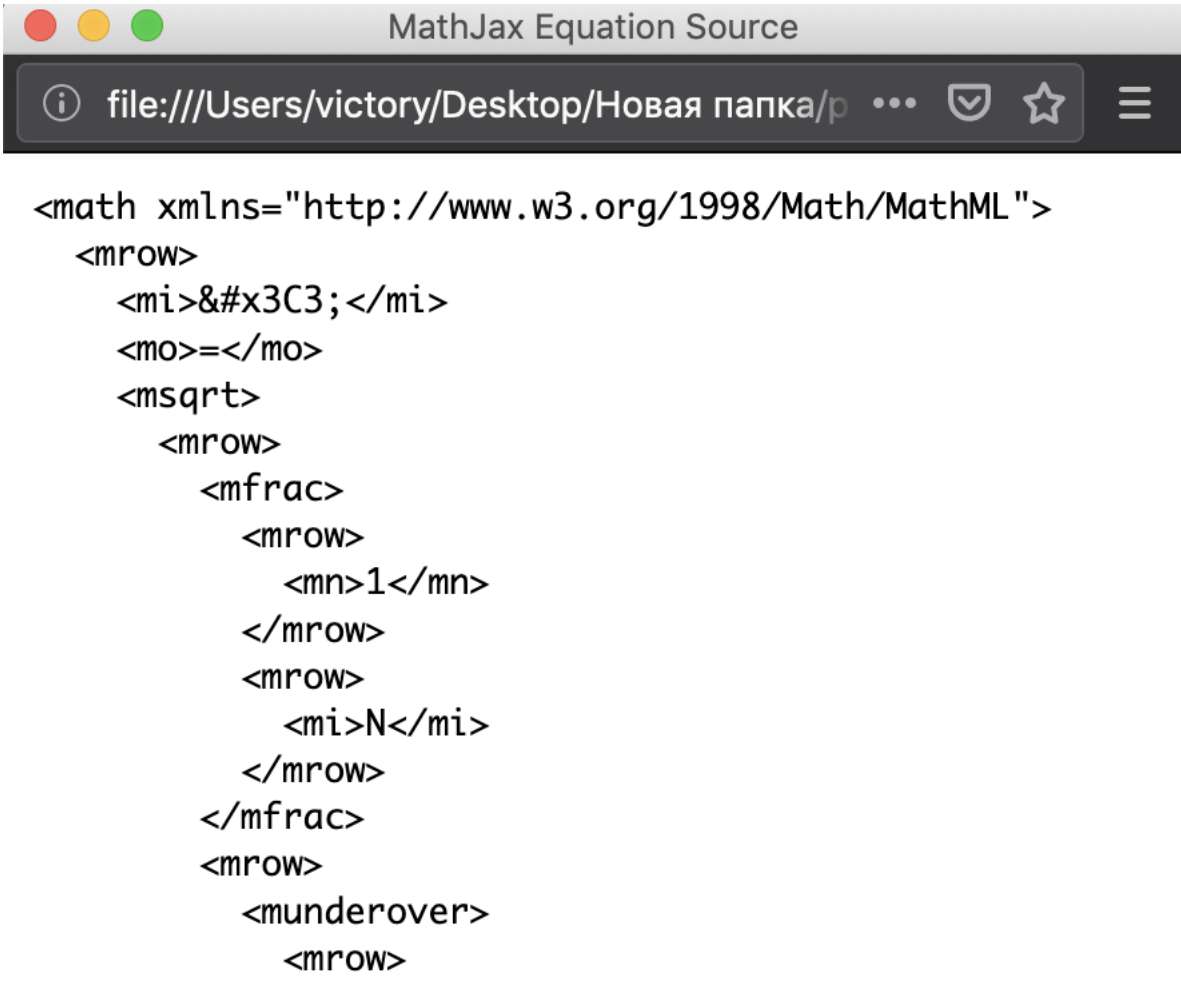


$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}.$$

Рисунок 3. Результат работы

Посмотрим структуру данного математического выражения с помощью MathML (см.Рисунок 4).

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$




```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi>&#x3C3;</mi>
    <mo>=</mo>
    <msqrt>
      <mrow>
        <mfrac>
          <mrow>
            <mn>1</mn>
          </mrow>
          <mrow>
            <mi>N</mi>
          </mrow>
        </mfrac>
        <mrow>
          <munderover>
            <mrow>
```

Рисунок 4. Просмотр структуры математической формулы

ЗАКЛЮЧЕНИЕ

В заключении работы можно отметить, что все поставленные задачи были выполнены.

Результаты решения задач:

1) Дано определение XSL – eXtensible Stylesheet Language – это расширяемый язык таблиц стилей – механизм управления отображением XML-документов, представляющий собой совокупность рекомендаций консорциума Всемирной паутины (W3C). Состоит из 2 частей (рекомендаций): XSLT и XPath.

2) Проанализированы XPath и XSLT, а также выявлена сущность связи между ними.

3) Рассмотрены основы MathML, а именно: способы представления математического материала и основные элементы MathML.

На данном этапе были изучены языки XSL и MathML в том объеме, который необходим для проектирования шаблона с использованием элементов математической разметки.

4) Спроектирован XSL шаблон.

Таким образом, цель данной работы – проектирование XSL-шаблона для визуального отображения структуры математического информационного ресурса – достигнута в полном объеме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) XSL. [Электронный ресурс] / Режим доступа: <https://www.w3.org/Style/XSL/>(дата обращения: 20.05.2019)
- 2) XSLT.Tutorial [Электронный ресурс] / Режим доступа: <https://msiter.ru/tutorials/xslt> (дата обращения: 20.05.2019)
- 3) XPath. Рекомендация W3C. [Электронный ресурс] / Режим доступа: <https://www.w3.org/TR/2014/REC-xpath-30-20140408/>(дата обращения: 20.05.2019)
- 4) XSL Languages [Электронный ресурс] / Режим доступа: https://www.w3bai.com/en-us/xsl/xsl_languages.html (дата обращения: 20.05.2019)
- 5) XPath Tutorial. [Электронный ресурс] / Режим доступа: https://www.w3schools.com/xml/xpath_intro.asp(дата обращения: 20.05.2019)
- 6) XSLT. Рекомендация W3C. [Электронный ресурс] / Режим доступа: <https://www.w3.org/TR/2017/REC-xslt-30-20170608/>(дата обращения: 20.05.2019)
- 7) Справочник по элементам XSLT. [Электронный ресурс] / Режим доступа: <http://xsltdev.ru> (дата обращения: 20.05.2019)
- 8) Технология XSLT. [Электронный ресурс] / Валиков А.Н. /Режим доступа: https://www.e-reading.life/bookreader.php/1016301/Valikov_-_Tehnologiya_XSLT.html (дата обращения: 21.05.2019)
- 9) Введение в XSLT, XPath. Лабораторная работа / [Электронный ресурс] / Государев И.Б. / Режим доступа: http://kodaktor.ru/xsl_lab.pdf (дата обращения: 29.04.2019)
- 10) Обзор инструментов для математических данных [Электронный ресурс] / Аджиев А.С., Атаева А.М., Шиолашвили Л.Н. /Вычислительный центр РАН /Режим доступа: http://rcdl.ru/doc/2005/sek8_4_paper.pdf (дата обращения: 22.05.2019)

11) Справочные данные по MathML[Электронный ресурс] / Режим доступа: <https://developer.mozilla.org/ru/docs/Web/MathML> (дата обращения: 22.05.2019)

12) Основы MathML. Представление математических текстов в Internet. Практическое руководство.[Текст]/ Елизаров А.М., Липачев Е.К., Малахальцев М.А. – Казань: Издательство Казанского математического общества, 2008. – 100 с.

13) XSLT MathML Library [Электронный ресурс] / Режим доступа: <http://sourceforge.net/projects/xshtml/> (дата обращения: 23.05.2019)

14) Руководства, инструменты и сообщества для разработчиков [Электронный ресурс] /Режим доступа: <https://www.ibm.com/developerworks/ru/>(дата обращения: 23.05.2019)