

- и опыт внедрения: сб. трудов Международного форума. Москва, 28–29 ноября 2019 г., МГТУ им. Н.Э. Баумана. М.: Изд-во МГТУ им. Н.Э. Баумана, 2020. С. 87–89.
- [4] Послание Президента РФ Федеральному Собранию от 01.03.2018 «Послание Президента Федеральному Собранию». URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_291976/](http://www.consultant.ru/document/cons_doc_LAW_291976/) (дата обращения 29.05.2021).
- [5] Федеральный закон от 25 декабря 2008 г. № 273-ФЗ (ред. от 31.07.2020) «О противодействии коррупции». URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_82959/](http://www.consultant.ru/document/cons_doc_LAW_82959/) (дата обращения 30.05.2021).
- [6] Официальный сайт новостного издания «Коммерсантъ» URL: <https://www.kommersant.ru/doc/3854957> (дата обращения 31.05.2021)
- [7] Мухамадиева Д.Н. Цифровизация против коррупции. Возникновение Е-коррупции // Вестник Самарского государственного экономического университета. 2019. № 1 (171). С. 9–13.
- [8] Федеральный закон от 09 февраля 2009 г. № 8-ФЗ (ред. от 30.04.2021) «Об обеспечении доступа к информации о деятельности государственных органов и органов местного самоуправления». URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_84602/](http://www.consultant.ru/document/cons_doc_LAW_84602/) (дата обращения 25.05.2021).
- [9] Федеральный закон от 22 декабря 2008 г. № 262-ФЗ (ред. от 08.12.2020) «Об обеспечении доступа к информации о деятельности судов в Российской Федерации». URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_82839/](http://www.consultant.ru/document/cons_doc_LAW_82839/) (дата обращения 25.05.2021).
- [10] Федеральный закон от 27 июля 2006 г. № 149-ФЗ (ред. от 09.03.2021) «Об информации, информационных технологиях и о защите информации». URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_61798/](http://www.consultant.ru/document/cons_doc_LAW_61798/) (дата обращения 27.05.2021).
- [11] Федеральный закон от 21 июля 2014 г. № 212-ФЗ (ред. от 27.12.2018) «Об основах общественного контроля в Российской Федерации». URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_165809/](http://www.consultant.ru/document/cons_doc_LAW_165809/) (дата обращения: 30.05.2021).
- [12] Овчинников А.И. Противодействие коррупции в условиях цифровизации: возможности, перспективы, риски // Журнал российского права. 2019. № 11. С. 158–170.
- [13] Никитина К.В., Буркацкая Н.В. Электронная Россия как способ борьбы с коррупцией // Современное общество и власть. 2018. № 3 (17). С. 261–265.

УДК 004.424.36

## Программное обеспечение для конвертации расчетных листингов MathCAD в MS Word

**Мамедов Владислав Марсельевич**

mamedov-vm@bk.ru

НИИ «Энергомашиностроение» МГТУ им. Н.Э. Баумана

**Архаров Иван Алексеевич**

20772@mail.ru

МГТУ им. Н.Э. Баумана

*Рассмотрены этапы разработки уникального программного продукта для конвертации расчетных листингов MathCAD в MS Word на серверном языке программирования PHP. Проведен анализ предварительных вариантов организации конвертации на уровне блочной модели, включающей системы подстановки численных значений и единиц измерения. На основании анализа выбран и реализован оптимальный вариант организации архитектуры программного комплекса, не имеющего аналогов по функциональности. Показаны преимущества предлагаемого подхода к организации системы конвертации.*

**Ключевые слова:** конвертация, рекурсия, алгоритм, MathCAD, MS Word, PHP, программа для ЭВМ

## Введение

Подавляющее большинство расчетов, проводимых в научно-технической сфере, выполняются в программных комплексах MathCAD, Maple и MatTab, поскольку они оснащены визуальной средой разработки с удобными инструментами для проведения сложных расчетов и мощными математическими ядрами. Наибольшую популярность в инженерной сфере имеет MathCAD благодаря низкому относительно конкурентов порогу вхождения для пользователя и возможностью подключать сторонние библиотеки без знания специфического синтаксиса среды.

С другой стороны, расчеты, выполненные в листингах MathCAD, не отвечают установленными ЕСКД требованиям к оформлению конструкторской документации. Возникает необходимость в дополнительном вспомогательном этапе разработки — оформлении результатов расчета в редактируемом формате MS Word или LaTeX. Цель настоящей статьи — осветить основные этапы разработки и оптимизации ПО для исключения низкоэффективного ручного оформления расчетов из технологической цепочки разработки конструкторской документации. Актуальность работы находит отражение в снижении трудоемкости и сокращении фонда времени на вспомогательные операции, что позволит снизить финансовые затраты на реализацию проекта [1].

## Постановка задачи

Процесс разработки программного продукта по конвертации расчетных листингов в редактируемый формат был начат с рассмотрения экономической эффективности. Затраты на вычислительные мощности системы, на которых реализуется работа программного продукта, и на разработку алгоритма в долгосрочной перспективе должны быть ниже, чем затраты на проведение ручной конвертации [2]. Поэтому к разрабатываемым алгоритмам предъявляются жесткие требования по следующим критериям:

- удобный и понятный интерфейс для конечного пользователя;
- минимальный расход системных ресурсов: памяти и системного времени CPU;
- скорость выполнения алгоритмов;
- возможность контроля содержимого передаваемых на конвертацию файлов и предотвращение критических ошибок в случае возникновения не предусмотренного исключения;
- минимальные ограничения на содержимое и оформление передаваемого на конвертацию файла;
- исключение человеческого фактора при использовании алгоритмов конвертации.

С учетом вышеизложенных требований, первоначально была рассмотрена возможность реализации программного продукта на серверном языке программирования. В этом случае исключается влияние объема располагаемых вычислительных ресурсов машины пользователя, поскольку работа алгоритма происходит на серверной стороне. Кроме того, пользователь всегда имеет доступ к последней версии программного продукта, что дает свободу действий при внедрении и развертывании новой версии алгоритмов конвертации. Помимо очевидных преимуществ применения серверного языка программирования, немаловажен негативный фактор: обязательное наличие подключения к сети Интернет на машине пользователя.

## Организация системы

Наиболее распространенный и доступный к развертыванию разрабатываемого продукта на серверной стороне язык программирования — PHP [3]. Для организации взаимодействия вычислительной машины пользователя (далее — клиента) с серверной частью предусматривается визуальный интерактивный интерфейс с применением full-stack технологий HTML (HyperText Markup Language) + CSS (Cascading Style Sheets) + JS (JavaScript) + AJAX (Asynchronous Javascript and XML) + PHP (Hypertext Preprocessor) [4–6]. Обмен данными между сервером и клиентом осуществляется в формате JSON (JavaScript Object Notation) [7]. В рамках пользовательского интерфейса предусматривается возможность загружать файл на конвертацию, отслеживать прогресс выполнения конвертации и загружать с сервера созданный алгоритмом файл результата. Благодаря ярко выраженной стратификации клиентской и серверной части, имеется возможность проводить разработку и доработку программного продукта параллельно с сохранением доступности и работоспособности продукта с высоким показателем up-time.

В процессе работы с программным продуктом пользователь взаимодействует с сервером напрямую единожды для получения содержимого страницы, а затем посредством AJAX-запросов (Asynchronous Javascript and XML). Данный факт позволяет классифицировать разрабатываемый программный продукт как полноценное веб-приложение.

При разработке программного продукта сначала составляется модульная блок-схема (рис. 1). Видно, что в процессе работы с приложением клиент передает и получает данные через отдельный модуль Data changer, в котором, кроме первичной проверки данных на стороне клиента, проводится дополнительная более глубокая верификация разрешения, типа и размера загружаемого файла, а также настроек, устанавливаемых пользователем для процесса конвертации.

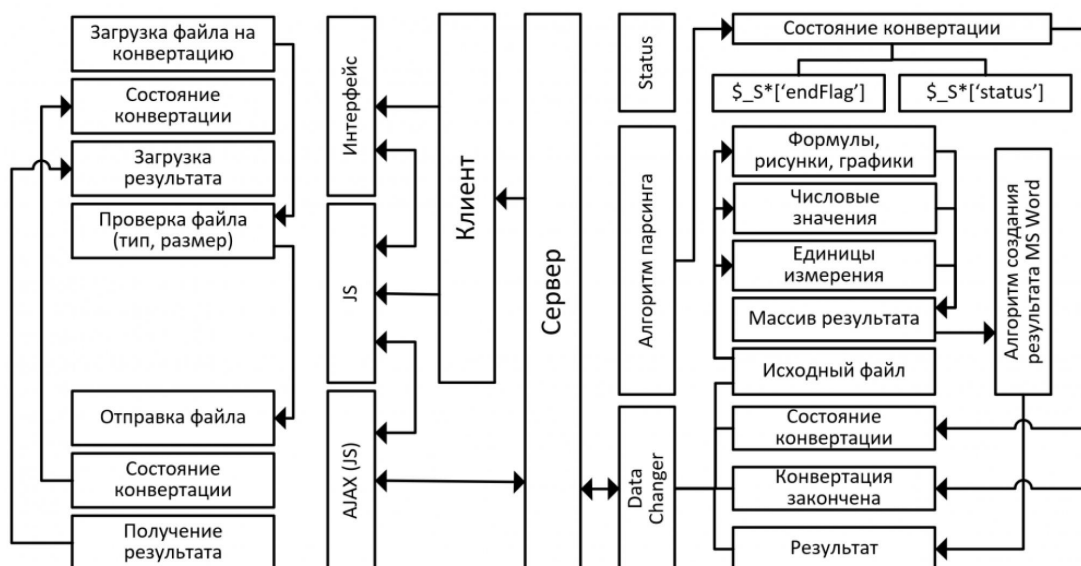


Рис. 1. Модульная блок-схема программного продукта

Управление состоянием выполнения конвертации (числовым и булевым) также вынесено в отдельный модуль Status, а передача данных осуществляется через суперглобальный массив `$_SESSION`. В результате такого подхода полностью исклю-

чена ситуация, в которой пользователь может получить критическую ошибку при выполнении алгоритмов конвертирования конкретного файла, и работоспособность приложения сохранится. Клиент запрашивает у сервера состояние числового и логического указателя с определенной периодичностью, выводя соответствующую информацию в окне интерфейса.

В научной сфере для типографического набора текстов, содержащих формулы, рисунки и графики, широкое распространение получил макропакет расширений LaTeX к системе компьютерной верстки TeX. LaTeX имеет компактную структуру вложенности и создаваемые с его помощью файлы имеют небольшой объем и могут быть переданы в виде текста, а затем скомпилированы на любой вычислительной машине [8]. Еще один положительный аспект этого формата заключается в возможности не только типографической печати, но и публикации расчетов в сети Интернет в электронном виде. Это становится возможным благодаря применению JS библиотеки MathJax для визуализации формул с поддержкой технологии TrueType в окне браузера большинства цифровых устройств. Поэтому изначально предполагалось выполнять конвертацию именно в этот редактируемый формат.

С другой стороны, самая популярная среда для подготовки, редактирования и передачи расчетной документации — MS Word. Для сохранения расширяемости и ее реализации в случае необходимости алгоритмы конвертации были построены на «словарях», добавление определенных правил конвертации в которые, делает возможным получение результата в любом требуемом формате. Словари включают шаблоны синтаксиса для конвертации операторов (39 шт.), символов и единиц измерения (122 шт.), реализованных в Mathcad. Пример операции в таком словаре показан на рис. 2.

```
<? php
...
$dictMML = array(
    ...
    'minus' => array(
        'operator' => '-',
        'pattern' => 's1<mo>-</mo>s2',
        'patterns' => array(
            0 => '/s1/',
            1 => '/s2/',
        ),
        'itemNum' => 2,
    ),
    ...
);
...
?>
```

**Рис. 2.** Пример операции вычитания в словаре MML

Формулы в MS Word набираемые посредством встроенного редактора, преобразуются к XML-структуре формата OMML (Office Mathematical Markup Language). Хранение данных в структуре XML (eXtensible Markup Language) приводит к значительному увеличению объема преобразованного файла за счет реализованного глубокого вложения и необходимости применения полноразмерных закрывающих тегов, в отличие от формата LaTeX.

Для составления словаря напрямую в OMML требуется значительно увеличить объем и сложность алгоритма, поэтому было принято решение применить промежуточный формат представления данных MML (MathML). В этом случае словарь синтаксиса операций имеет сравнительно небольшой объем, а алгоритм легко читаем и компактен. Для приведения промежуточного формата MML к OMML применяется поставляемый вместе с продуктами MS Office XSL (eXtensible Stylesheet Language) [9] преобразователь и визуализатор XML документов MML2OMML.XSL. Конвертация в формат LaTeX выполняется без применения вспомогательного представления данных.

Промежуточное преобразование привело к увеличению времени обработки конвертируемого файла, но за счет значительного уменьшения объема программного кода и специализированного преобразователя, разработанного Microsoft, общая отказоустойчивость системы возросла.

Структура MathCAD файлов 14 и 15 версии также имеет XML формат. Поскольку не существует XSL транслятора между форматами MathCAD в MML, алгоритм преобразования сводится к прямому парсингу XML структуры и поэлементному разбору. Для этого исходный Mathcad файл после загрузки на сервер и начала работы алгоритма предварительно очищается от метаданных и системной информации, которые значительно загружают оперативную память серверного оборудования.

Для полноценной работы алгоритма конвертации необходимо изменить системную переменную `max_execution_time` PHP сервера, ограничивающую максимальное время выполнения скрипта, в файле `php.ini` на значение не более 120 (задается в секундах). Полное снятие ограничения на время выполнения скрипта позволяет конвертировать файлы с большим содержанием, но в случае критической ошибки приводит к созданию процесса, который не будет завершен сервером до перезагрузки, что недопустимо в рамках установленных ограничений.

Ядро алгоритма основано на рекурсивном вызове основной функции парсинга с передачей последующей вложенной структуры. В общей сложности в структуре листинга Mathcad имеется пять наиболее значимых примитивов, к которым относятся формулы (MATH), тексты (TEXT), рисунки (PICTURE), графики (PLOT) и области (AREA). В рамках разрабатываемого приложения принято решение учитывать только первые четыре, а области (AREA) оставить для возможности скрытия пользователем той части расчета, конвертация которой в MS Word не требуется.

Как видно из схемы (рис. 3), XML структура MathCAD состоит из регионов, которые идут в порядке их визуального размещения в листинге [10]. Для каждого региона вызывается метод `Parse()`, в который передается указатель на текущую рассматриваемую структуру. Этот метод снова обращается к XML и получает информацию о вложенной структуре и производит ее анализ.

Различается несколько подтипов структуры MATH: программы, формулы, уравнения и переменные. Подструктура MATH ID отвечает за переменные, определенные в листинге, поэтому при обнаружении парсером новой переменной вызывается метод `InitializeVar()`, который проверяет численное значение и размерность вновь определенной переменной и записывает в массив `VarArr[]` указатель на эту переменную и ее свойства. Аналогичный метод `InitializeFunc()` вызывается для подтипа MATH DEFINE, задача которого проверить и записать указатель на функцию и ее свойства.

В MathCAD 14 и 15 версии нет возможности переопределить размерности физических величин в выражении при их подстановке методом `explicit` в выражение, поэтому в разработанном алгоритме применяется специальный словарь `UnitArr[]`, в котором содержится информация о размерности и шаблон для замены латинских обозначений на кириллические в Международной системе единиц (СИ). Если же в расчетном листинге не применялся метод `explicit`, то можно включить режим подстанов-

ки числовых значений и единиц измерения с необходимой точностью в настройках самого приложения. Благодаря этому возможно снижение затрат на простановку метода explicit по всем выражениям в рамках расчетного листинга MathCAD.

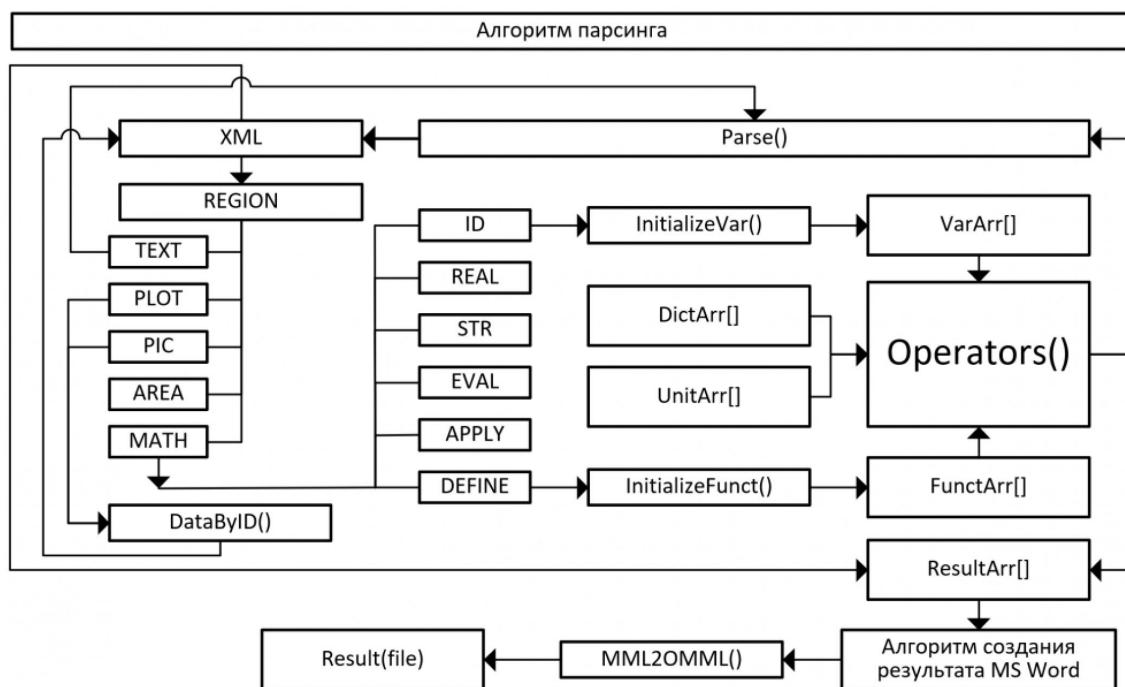


Рис. 3. Увеличенная схема алгоритма парсинга структуры MathCAD xml

Рисунки и двумерные графики переносятся в MS Word путем извлечения изображения из XML структуры в кодировке base-64 по указателю, содержащемуся в свойствах региона PLOT и PICTURE, и вставки этого изображения в документ Word как HTML-элемент. Возможность переноса трехмерных графиков остается перспективной к разработке, но имеются сложности преобразования массива данных с последним кадром графика в HTML-элемент.

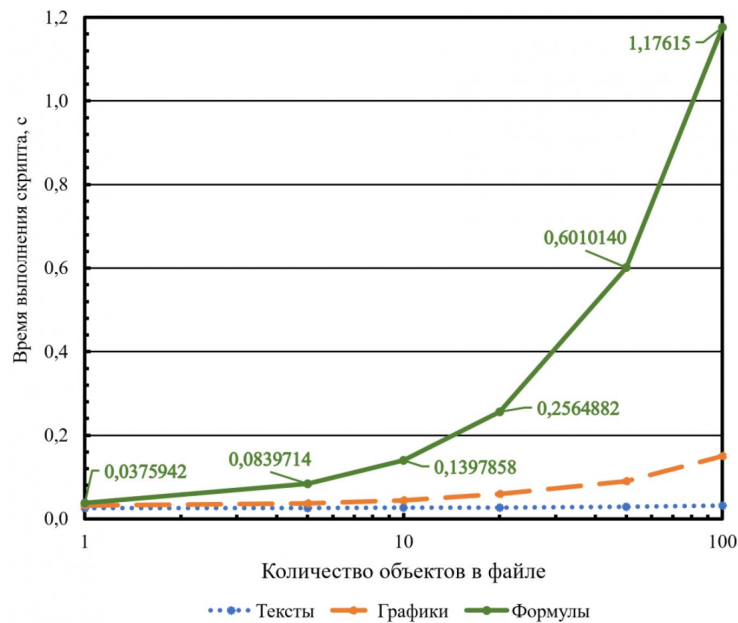
Разработанный программный продукт [11] успешно обрабатывает подпрограммы, содержащие логические, арифметические и другие операторы; матрицы и различные действия с ними; векторы-строки и столбцы. На ряду с достигнутыми результатами остается открытым вопрос переноса текста, содержащего формулы. Проблема возникает из-за смещения текста и формул в формате OMML, при обработке которого MS Word перестает считывать и визуально отображать структуру документа. Текст, не содержащий формул, переносится без возникновения ошибок.

## Результаты

Для визуального представления быстродействия работы разработанного алгоритма парсинга были проведены тестовые конвертации. Конфигурация сервера:

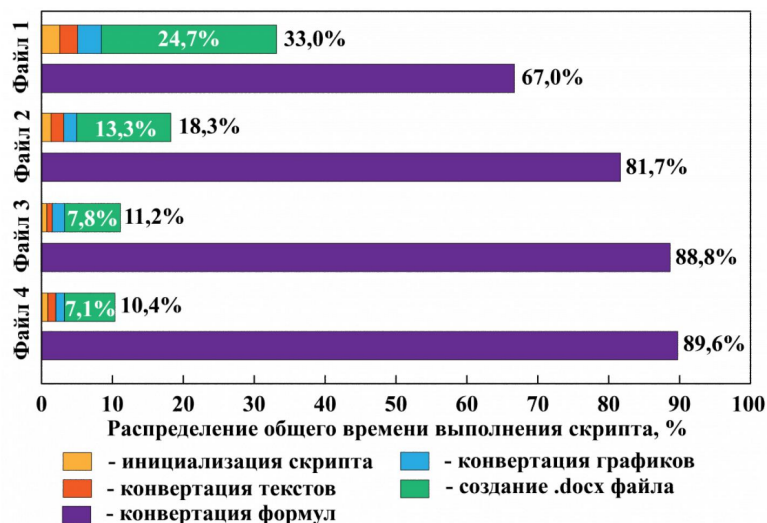
- Open Server: x64 5.2.2.0;
- версия PHP: 7.3-x64;
- версия HTTP: Apache 2.4;
- ОЗУ: 15,95 Гб 3000 МГц;
- ОЗУ, занятое системой: 6,03 Гб;
- процессор: AMD Ryzen 7 1700 3,90 ГГц.

На графике (рис. 4) представлены результаты замеров системного времени выполнения скрипта с помощью функции `microtime()` [12], как среднее значение между пятью последовательно проведенными конвертациями.



**Рис. 4.** Быстродействие алгоритма конвертации при количественном и качественном изменении содержимого

Из графика (см. рис. 4) было получено усредненное время непосредственно инициализации RHP скрипта, которое составляет 0,028 с. Помимо инициализации, время затрачивается на создание MS Word файла из подготовленного массива данных при помощи библиотеки `rhword` (версия 0.17.0, подключение зависимостей выполнено через `composer`). Для анализа распределения системного времени были проведены замеры при конвертации файлов (характеристика файлов и время выполнения конвертации приведены в таблице), результаты которых представлены графически (рис. 5). Для расчета времени конвертации формул, графических и текстовых блоков были использованы зависимости (см. рис. 4).



**Рис. 5.** Распределение временных затрат по процессам в ходе конвертации

Проводя анализ распределения временных затрат, можно установить, что на создание результирующего файла MS Word библиотекой `phpword` затрачивается в среднем 10...15 % от общего времени конвертации. Значительная часть этого времени идет на подключение и инициализацию самой библиотеки. Отсюда следует преимущество формата LaTeX, состоящее в том, что для получения конечного результата не требуется время на работу вспомогательной библиотеки. С учетом значительного снижения информационного объема и отображения результата конвертации на экранах большинства современных устройств формат LaTeX в большей степени удобен для хранения в текстовых полях баз данных веб-приложений и информационных технических ресурсов.

#### Результаты замера времени выполнения скрипта для файлов произвольного содержания

Содержание	Файл			
	1	2	3	4
Регионы	64	295	348	374
Графики	3	3	27	6
Формулы	44	139	266	252
Тексты	26	157	0	159
Строк подпрограмм	15	0	4	0
Время выполнения скрипта, с	1,056058	1,988231	3,476713	3,25962

Помимо измерения времени выполнения скрипта были проведены замеры использования оперативной памяти. Для простых задач по конвертации текстов и формул используется не более 2 Мб памяти, а для высоконагруженных и разнообразных по содержанию файлов не более 6 Мб. На основании полученных результатов вариант организации программного комплекса на серверном языке может быть развернут на машине с ограниченными вычислительными ресурсами с учетом небольшого времени выполнения скрипта и потребления памяти.

## Заключение

Было разработано уникальное программное обеспечение для конвертации файлов MathCAD в MS Word и LaTeX, рассмотрены различные варианты реализации архитектуры приложения. Благодаря принятым решениям была достигнута высокая производительность и безотказность алгоритма парсинга MathCAD файлов с сохранением оригинальной структуры конвертируемых файлов в конечном представлении, возможность подставлять единицы измерения и численные значения в указанном пользователем формате. Разработанное приложение имеет пути дальнейшего развития и совершенствования как по клиентской, так и по серверной части, однако уже в текущей редакции продукт не имеет аналогов по функциональности, позволяет создавать полностью адаптированные, удобные к передаче и публикации результаты конструкторских расчетов.

На фоне быстрого развития веб-технологий и автоматизации проектирования разработанное приложение планируется внедрить в проектно-конструкторскую деятельность студентов и на основании реального опыта использования доработать программный код для дальнейшего предложения к применению на научно-производственных предприятиях.



## Литература

- [1] Организация производства и управления предприятием / О.Г. Туровец, М.И. Бухалков, В.Б. Родинов и [др]. 2-е изд. М.: ИНФРА-М, 2009. 544 с.
- [2] Липсиц И.В., Коссов В.В. Экономический анализ реальных инвестиций. 2-е изд., перераб. и доп. М.: Экономистъ, 2004. 347 с
- [3] Котеров Д.В., Симдянов И.В. PHP 7. СПб.: БХВ-Петербург, 2016. 1088 с.
- [4] Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов / пер. с англ. М.А. Райтмана. М.: Эксмо, 2013. 480 с.
- [5] Флэнаган Д. JavaScript. Подробное руководство. СПб: Символ-Плюс, 2008. 992 с.
- [6] How to use \$.ajax(). URL: <https://dev.to/williamragstad/how-to-use-ajax-3b5e> (дата обращения 19.10.2020).
- [7] Сергушкин В.В., Бодров О.А. Использование формата данных JSON для взаимодействий с пользователем при разработке программного обеспечения аппаратуры передачи данных // Современные технологии в науке и образовании – 2018: матер. Междунар. науч.-технич. форума. В 10 т. Рязань, 28 февраля – 2 марта 2018, Рязан. гос. радиотехн. ун-т. Рязань: Рязан. гос. радиотехн. ун-т, 2018. Т. 5. С. 217–220.
- [8] Котельников И.А., Чеботаев П.З. LaTeX по-русски. Новосибирск: Сибирский хронограф, 2004. 489 с.
- [9] XML Tutorial. Volume 8: The XSLT Stylesheet and XPath. URL: <http://www.xmlmaster.org/en/article/d01/c08/> (дата обращения 19.10.2020).
- [10] Очков В.Ф. Mathcad 14 для студентов и инженеров: русская версия. СПб.: БХВ-Петербург, 2009. 512 с.
- [11] Свидетельство о государственной регистрации программы для ЭВМ №2021617133. Программа для конвертации файлов из XMCD в DOCX MathEssence / В.М. Мамедов. Заявка № 2021616315. Дата поступления 27 апреля 2021 г. Зарегистрировано в Реестре программ для ЭВМ 11 мая 2021 г.
- [12] Официальное руководство по PHP. URL: <https://www.php.net/manual/ru/> (дата обращения 19.10.2020).

УДК 681.513

## Компенсация отраженных сигналов спутниковой навигационной системы

**Масленников Андрей Леонидович**  
МГТУ им. Н.Э. Баумана

amas@bmstu.ru

*Рассмотрены факторы, вызывающие погрешности навигационной информации при работе спутниковых навигационных систем. Исследованы погрешности, обусловленные многолучевой интерференцией. Представлены способы компенсации погрешностей спутниковой навигационной системы, позволяющие компенсировать влияние возмущающих факторов, в частности погрешностей, вызванных радиосигналами, отраженными от препятствий. Предложено построить модели отраженных сигналов с помощью эволюционных алгоритмов.*