

Технологии документирования научного контента. II. Электронная книга

В. А. Нечитайленко

Геофизический центр РАН, Москва, Россия

Abstract. Статья посвящена проблеме построения конвертера для преобразования \LaTeX документов в HTML5 и EPUB3. Суть предложенного и реализованного автором решения заключается в замене макроопределений используемого \LaTeX класса новыми макроопределениями, позволяющими эмулировать XML-совместимую структуру документа. Набор таких макроопределений представляет конечное множество, что позволяет построить алгоритм конверсии \LaTeX в XML/XHTML. Хотя предложенное решение и ограничивает возможности \LaTeX , оно вполне укладывается в стандарты представлений, определенных издателем для конкретного журнала.

Введение

Начиная с конца 80-х годов прошлого столетия язык разметки научных текстов \LaTeX получил широкое распространение и развитие и стал стандартом *де-факто* в научной среде, в особенности в области точных и технических наук, наук о Земле и др. Выбор \LaTeX в качестве основного первичного языка разметки был предопределен отсутствием реальных конкурентов, особенно в случае сложных математических и химических текстов.

С развитием телекоммуникаций и появлением первых стандартов языков представления текстов в компьютерных сетях сразу же встал вопрос о необходимости конверсии \LaTeX -текстов в HTML и позднее в XML. В общем случае корректный перевод произвольных \LaTeX -текстов в XML невозможен, поскольку $\text{\TeX}/\text{\LaTeX}$ несовместим с концепцией SGML.

Именно поэтому все известные разработки конвертеров текстов в \LaTeX к HTML текстам ограничены фиксированным списком обрабатываемых макроопределений \LaTeX . Дополнение же указанных конвертеров механизмами расширения этих списков, путем добавления в программы соответствующих модулей, не устраняет проблему, хотя и весьма полезно для адаптации конвертера к решению конкретной задачи (т.е. конверсии документа с заданным DTD).

Очень важным является понимание принципиального различия между концепцией XML как подмножества SGML, которая определяет объекты, составляющие документ, их структуру и свойства, не заботясь о том, как документ представлен на физическом уровне (носителе), и концепцией \TeX , которая сводится по сути к описанию поведения пера на бумаге, не более того.

Распространенное представление о том, что макропакеты, такие как \LaTeX , \AmSTeX и др. преодолевают это различие, – не более чем заблуждение. В работе известных \TeX guru Миттельбаха и Роули [1997,

р. 196] читаем:¹

Поскольку типичное SGML DTD использует концепцию подобную концепции L^AT_EX, форматирование часто реализуется простой заменой элементов документа конструкциями L^AT_EX, вместо того, чтобы использовать непосредственно примитивы T_EX'а. Это позволяет использовать сложные аналитические методы, встроенные в L^AT_EX, и избежать необходимости программировать в T_EX.

Упомянутое в этой цитате подобие концепций является не прямым, а лишь опосредованным, т.е. не существует 1-1 соответствия.

Для полноты картины процитирую ответ еще одного T_EXguru [*Sebastian Rahtz*, 2005; <http://www.tug.org/pipermail/tex-live/2005-March/007846.html>] на вопрос одного из пользователей, можно ли найти хороший

¹Because a typical SGML Document Type Definition (DTD) uses concepts similar to those of L^AT_EX, the formatting is often implemented by simply mapping document elements to L^AT_EX constructs rather than directly to 'raw T_EX'. This enables the sophisticated analytical techniques built into the L^AT_EX software to be exploited; and it avoids the need to program in T_EX.

инструмент для конверсии \LaTeX файлов в XML:²

Существует несколько инструментов, ни один из которых нельзя легко использовать или гарантировать удовлетворительные результаты. Конверсия \LaTeX к XML от природы чрезвычайно сложна, чтобы быть 100% правильной. По моему мнению это плохая идея :-)

От плохой идеи к прагматичному решению

Документ (научная статья) включает множество объектов, которые несут большую или меньшую семантическую нагрузку. Для лучшего восприятия эти объекты (заголовки разного уровня, фигуры, таблицы, списки, в том числе реферативные, текстовые выделения, гипертекстовые ссылки и пр.) каким-то образом акцентируются, правила акцентирования определяются стандартами издателя и культурной традици-

² ... there are several, none of them easy to use or guaranteed to give satisfactory results. \LaTeX to XML is, inherently, extraordinarily hard to 100% right. In my personal view, it's a Bad Idea :-)

ей и могут быть одинаковыми для разных объектов или разными для одного класса объектов.

Следует ли из этого, что для генерации версий документов, например, в PDF/PS и в HTML/XML нужно иметь различные соответствующим образом сформированные исходные файлы?

Да, если мы хотим использовать всю мощь издательской системы $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ с ее многочисленными расширениями и возможностью включения авторских макроопределений непосредственно в исходный файл, а также включить в документ динамические и интерактивные объекты.

Нет, если издатель и автор готовы использовать ограниченный перечень макроопределений, которые, с одной стороны, определяют правила интерпретации исходного текста для $\text{T}_{\text{E}}\text{X}$ engine и, с другой стороны, эмулируют XML совместимую структуру, обеспечивая таким образом возможность программной конверсии исходного файла к форматам HTML5/XHTML/EPUB3.

Использование языков типа XML в принципе позволяет не только учитывать специфику различных предметных областей, но и повышать эффективность структурирования при автоматизированной обработке статей и иных документов для последующей ин-

дексации. Однако без полного онтологического описания предметной области использование XML для семантического структурирования так же ограничено, как и использование L^AT_EX, который, как уже отмечалось, являясь языком представления, содержит семантику лишь опосредованно.

Издаваемые сегодня электронные журналы, равно как и бумажные, ориентируются, в первую очередь, на читателя, а не на машину. В то же время есть все основания полагать, что в обозримом будущем научные публикации будут ориентироваться, в первую очередь, на ввод в информационно-поисковые (интеллектуальные) системы с развитым межмашинным интерфейсом.

А пока... лучшее решение – следовать рекомендации, прозвучавшей на Первой конференции экспертов ЮНЕСКО и МСН по электронным публикациям в науке (Париж, 1996): “Сделай, потом фиксируй!”³

Именно такой подход был реализован автором при разработке программного пакета ELXpaper (ELectronic eXtended paper) для изданий, публикуемых ГЦ РАН (*Russian Journal of Earth Sciences, Вестник Отделения наук о Земле РАН и Geoinformatics Research Papers*).

³Do it, then fix it! Из коллекции утверждений, собранных в работе [Arnoud de Kemp, 1996].

Программный пакет ELXpaper

Пакет включает три основных компонента: стилевой файл `elxpaper.sty`, Perl скрипт `elxpaper.pl` с конфигурационным файлом `elxpaper.cfg`, Perl скрипт `elxtohtml.pl` с конфигурационным файлом `elxtohtml.cfg`, а также набор основных предварительно созданных элементов (CSS файлы, логотипы и макеты обложек и титульных страниц публикуемых изданий и т.п.).

Стилевой файл `elxpaper.sty` является расширением стандартного класса \LaTeX `article.cls` и поддерживает двухколонный журнальный формат, генерацию внутренних и внешних активных гиперссылок, генерацию сообщений об ошибках в исходном файле или предупреждений, генерацию результата трансляции в форматах DVI или PDF, а также генерацию XML-метаописаний публикуемых документов для регистрации в системе CrossRef (в соответствии с XML-схемой CrossRef 4.3.3) и загрузки в Научную электронную библиотеку eLIBRARY.RU (в соответствии с XML-схемой eLibrary CE 7.1.4.1284).

Последняя задача решается непосредственно в процессе трансляции исходного текста благодаря тому обстоятельству, что $\text{\TeX}/\text{\LaTeX}$ – это не просто один

из языков разметки текста, а, по существу, программная система, реализующая концепцию программирования машины Тьюринга. Конечно, возможности программирования в $\text{T}\text{E}\text{X}/\text{L}\text{A}\text{T}\text{E}\text{X}$ существенно уступают возможностям языков программирования высокого уровня, но они оказываются достаточными для построения множества макроопределений, расширяющих стандартные классы $\text{L}\text{A}\text{T}\text{E}\text{X}$ и эмулирующих семантическую структуру документа. Детали работы `e1xpaper.sty` рассмотрены в работе [Нечитайленко, 2015]

Perl скрипт `e1xpaper.pl` транслирует исходный $\text{L}\text{A}\text{T}\text{E}\text{X}$ файл в набор HTML5 файлов с генерацией основного файла с динамическим оглавлением и набора файлов, вызываемых из оглавления и из текста статьи, включающих фигуры, таблицы, и ряд вспомогательных файлов. Трансляция является многопроходной и включает следующие шаги:

- выделение фрагментов для последующего формирования из них иконок (для фигур и таблиц);
- нумерация строк исходного $\text{L}\text{A}\text{T}\text{E}\text{X}$ кода и слияние их в параграфы;
- разделение параграфов на семантически значимые фрагменты (например, если имеется дисплейный элемент внутри параграфа);

- обработка преамбулы с фиксацией основных параметров;
- выделение множества защищенных фрагментов (включая *verbatim* блоки и элементы) для последующей подстановки шрифтов, замены диакритических символов символами UTF-8, подстановки внутрискриптовой математики и разметки шрифтов;
- извлечение из текста элементов оглавления, метаданных и реферативных списков;
- генерация XML метаданных для CrossRef и eLibrary
- обработка внутренних и внешних гиперссылок;
- извлечение из исходного документа плавающих элементов (фигур и таблиц), подстраничных ссылок и табулированных фрагментов;
- конверсия основного документа к формату HTML5;
- конверсия дополнительных файлов (фигур, таблиц, оглавления, карты статьи и др.);
- генерация пакетного файла, содержащего набор команд для перевода набора HTML файлов в набор XHTML с последующим циклическим вызовом программы `elxhtml.pl` (см. далее);
- генерация структуры XHTML файлов, предусмотренных стандартом EPUB3, включая TOCXXX, `package.opf` и др.;

- генерация титульной страницы и обложки для электронной книги в формате EPUB3 на основе предустановленных шаблонов и специфической информации конкретного документа (статьи).

Конфигурационный файл `elxpaper.cfg` содержит описания параметров всех обрабатываемых в `elxpaper.pl` команд, включая их маски (patterns), регистрационные данные журналов, публикуемых ГЦ РАН, команды вызова вспомогательных free-domain программ и др. Кроме стандартных команд $\text{\LaTeX}2_{\epsilon}$, определенных в `article.cls` (некоторые из них переопределены, например, `title`, `abstract` и др.), включен также ряд команд, определенных в пакетах, расширяющих `article.cls`, в первую очередь в пакете `hyperref.sty`. Перечень используемых пакетов \LaTeX включен в преамбулу `elxpaper.sty`.

Скрипт `elxpaper.pl` работает в предположении, что (i) исходный файл статьи в \LaTeX не порождает сообщений об ошибках при его трансляции драйверами `latex` или `pdflatex`, (ii) в текущей директории присутствует файл `jobname.aux`, (iii) в преамбуле и тексте статьи не содержатся авторские макроопределения, формируемые командами типа `\newcommand`, `\renewcommand`, `\def`, `\gdef`, и т.п.

Все команды в тексте статьи, не имеющие семантической нагрузки и используемые только для форматирования текста, такие, например, как `\space`, `\skip`, `linebreak`, и т.п. игнорируются, равно как команды, имеющие семантическую нагрузку, но для которых в скрипте не определен алгоритм их обработки. Команды управления шрифтами заменяются соответствующими определениями в CSS файле `elx-online.css`.

Perl скрипт `elxtomml.pl` преобразует сгенерированные на предыдущем шаге файлы в HTML5 формате в файлы формата XHTML, что необходимо для включения последних в структуру EPUB3. При этом некоторые элементы конструкции исходного файла заменяются на более строгие конструкции XHTML. При публикации документов на русском языке осуществляется также перевод исходного текста в Юникод.

Скрипт `elxtomml.pl` решает также задачу преобразования математических и химических формул (как дисплейных, так и встроенных непосредственно в параграфы текста) представленных в нотификациях L^AT_EX к формату MathML. Как известно, издательская система T_EX/L^AT_EX/A_msT_EX обеспечивает своего рода стандарт *де-факто* для представления

математики при документировании научного контента. С переходом от POP (printed-on-paper) публикаций к электронным публикациям возникла необходимость найти способ представления $\text{T}_{\text{E}}\text{X}$ математики в форму, которая могла быть отражена в окне веб-браузера. В первых реализациях подобных решений широко использовалась замена математических и химических формул изображениями. Такое решение хотя и могло быть реализовано достаточно простыми средствами, но имело ряд серьезных недостатков, основными из которых стали проблема масштабирования изображений и проблема интерпретации их поисковыми системами.

С разработкой языка разметки математического текста *MathML*, который стал частью официально рекомендованного Консорциумом W3C стандарта HTML5 и в 2015 зарегистрирован в качестве стандарта Международной организации стандартов (ISO/IEC DIS 40314), задача свелась к конверсии математики, представленной в $\text{T}_{\text{E}}\text{X}/\text{A}_{\text{M}}\text{S}\text{T}_{\text{E}}\text{X}$ в формат *MathML*. В настоящее время среди многочисленных решений этой задачи особое место занимает система *MathJax*, разработанная и поддерживаемая *MathJax* консорциумом – совместным предприятием Американского математического общества (AMS) и Общества индустрии

стриальной и прикладной математики (SIAM).

MathJax не является традиционным конвертером файлов одного формата в файлы другого. Конверсия осуществляется “на лету” при загрузке в браузер HTML файла, имеющего в теле (`<body>`) математические фрагменты в $\text{T}_{\text{E}}\text{X}$ нотациях, а в преамбуле (`<head>`) команды вызова MathJax скриптов либо из сети серверов MathJax (<https://www.mathjax.org/>), либо из отдельной копии MathJax, устанавливаемой на пользовательском сервере. MathJax использует CSS с веб-шрифтами или SVG, вместо рисунков (bitmap или Flash), благодаря чему математика масштабируется одновременно с текстом при всех уровнях увеличения/уменьшения. При этом результат трансляции направляется непосредственно на экран.

В пакете ELXraper конверсия математических текстов для HTML5 и EPUB3 реализована двумя различными способами. Для онлайн-версии статьи в формате HTML5 достаточно сохранить математику в том виде, как она представлена в исходном $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ файле, добавив в преамбулу (`<head>`) вызов MathJax скриптов.

Заметим, что в реализованной нами версии пакета ELXraper MathJax скрипты вызываются непосредственно с сайтов сети cdn.mathjax.org. Для этого

рабочий компьютер должен быть подключен в Интернету. Возможна также установка MathJax на локальном сервере (см. детали). Второе решение в ряде случаев может оказаться более предпочтительным, но оно потребует регулярного обновления локальной версии MathJax, тогда как непосредственное соединение с MathJax CDN избавляет от этой необходимости.

Для электронной книги в формате EPUB3 такое решение непригодно, так как электронная книга в соответствии с рекомендациями IDPF (International Digital Publishing Forum) должна обеспечивать корректную интерпретацию текстов при отсутствии доступа к сети (offline). Устройства чтения документов в формате EPUB3, равно как эмуляторы таких устройств на десктопах, в соответствии с рекомендациями IDPF должны обеспечивать корректную интерпретацию HTML5/ XHTML, составной частью которых является MathML.

Как известно, формат EPUB представляет собой сжатый архив всех файлов, составляющих публикацию и отвечающих стандарту XHTML, с добавлением файлов, определяющих структуру архива (package.opf) и его тип (mimetype). Т.е. при переводе HTML5 файлов в более строгий формат

XHTML необходимо конвертировать математические $\text{T}_\text{E}\text{X}$ фрагменты (дисплейные и строчные) во фрагменты MathML.

Анализ существующих конвертеров файлов одного формата в файлы другого показал, что по многим параметрам они не могут конкурировать с системой MathJax (ограничения на исходные файлы, отсутствие поддержки, коммерческие условия использования и т.п.) В то же время MathJax, хотя и создаст MathML фрагмент для каждого $\text{T}_\text{E}\text{X}$ фрагмента с возможностью их просмотра во всплывающих окнах, но не позволяет вывести их в отдельный файл, что связано с ограничениями языка JavaScript.

Нами найдено и реализовано решение, которое по выражению автора $\text{T}_\text{E}\text{X}$ Дональда Кнута может быть отнесено к категории “dirty tricks”. Суть его в том, что из JavaScript мы не можем записать сформированный MathML файл на локальном компьютере, но можем послать его в качестве содержимого HTML формы на сервер, содержащий программу обработки этой формы с записью файла во внешнюю память. Наиболее просто это можно сделать установив такой специальный веб-сервер непосредственно на localhost, где установлен пакет ELXraper. Тогда результат обработки на этом сервере будет очевид-

ным образом доступен соответствующей программе пакета. Вопрос синхронизации работы этих программ решается вполне ordinarily через систему “флагов”. Если по каким-то причинам установка такого вспомогательного сервера на localhost невозможна или нежелательна, например при использовании разных компьютеров для подготовки статей к публикации, соответствующие скрипты и директорию для записи конвертированных файлов можно установить на любом доступном веб-сервере.

Конфигурационный файл `elxtomml.cfg` определяет порядок связи `elxtomml.pl` со вспомогательными программами (Рис. 1) взаимодействия с системой MathJax, включая адреса директорий `doc` и `cgi-bin` вспомогательного сервера, тип используемого в процессе конверсии браузера и команды его открытия и закрытия, адреса служебных файлов формируемого EPUB3 архива.

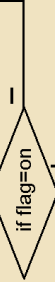
Дополнительные компоненты пакета представляют набор основных предварительно созданных элементов (рисунки логотипов и макеты обложек, CSS файлы, таблицы соответствия диакритических символов \LaTeX символам UNICODE, и др.) Перечень вспомогательных программ и порядок установки и

Скрипт *elxtohtml.pl*

1. Копирование мат. фрагментов и замена их уникальными маркерами (файл `jobname_tex.html`)
2. Заполнение формы выделенными мат. фрагментами для последующей конверсии (`jobname_mml.html`)
3. Модификация MathJax скрипта для накопления мат. фрагментов в JS форме

Отправка `jobname_mml` на сервер

Циклический опрос состояния флага

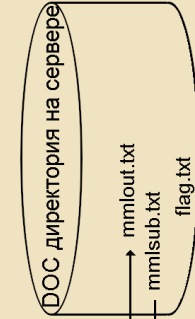


Чтение фрагментов конвертированных к MathML

Замена маркеров конвертированными в MathML фрагментами

Скрипт *textomml.pl*

Начальное состояние `flag=off`



1. Чтение содержимого формы
2. Склеивание фрагментов строк
3. Удаление встроенных комментариев
4. Замена маркеров номерами формул
5. Установка состояния `flag=on`

Установка состояния `flag=off`

Рис. 1. Схема взаимодействия скриптов `elxtohtml.pl` и `textomml.pl` в рабочей директории и в DOC директории вспомогательного сервера.

использования пакета ELXraper включены в Приложение.

Заключение

Разработанная технология успешно используется в практике редакционной подготовки изданий Геофизического центра РАН [Астапенкова и др. 2015], в первую очередь Russian Journal of Earth Sciences. Каждая из опубликованных в RJES статей, начиная с июня 2012 г., представлена в четырех форматах: (i) основная версия (version of record) в формате PDF; (ii) версия в формате HTML5; версия в формате EPUB3; а также (iv) версия в формате PDF, адаптированная для малых экранов (е-ридеров и смартфонов).

Следует иметь в виду, что на момент написания этой статьи большая часть устройств и программ чтения документов в формате EPUB3 не обеспечивает корректного воспроизведения, включая, в том числе, и Readium, рекомендованный IDPF в качестве основы для включения в ПО для планшетов, смартфонов и электронных книг. Наши тесты показали, что лучшие результаты обеспечивают приложение iBooks2 на iPad и iPhone, Gyan ePub Reader и некоторые эму-

ляторы устройств чтения электронных книг, в частности, AZARDI и EPUBReader: Add-on for Firefox.

Все сгенерированные версии в EPUB3 формате подвергаются проверке на соответствие стандарту с использованием программы epubcheck3.0 и рекомендованной IDPF программы EPUB Validator, после чего загружаются на сервер при отсутствии сообщений об ошибках и других типах предупреждений.

Благодарность. Автор признателен своим коллегам А. А. Астапенковой и Э. О. Кедрову за ценные критические замечания и помощь в подготовке статьи к печати.

Приложение

Описанный здесь пакет доступен в виде архива elxraper_1.5.zip на сайте электронных публикаций ГЦ РАН. В архиве файл ReadMeFirst_rus.pdf содержит всю необходимую информацию по установке и использованию пакета. Архив включает также исходную версию этой статьи в \LaTeX формате, из которой после обработки пакетом ELXraper были получены HTML и EPUB3 версии данной статьи.

Литература

- Астапенкова, А. А., Э. О. Кедров, В. А. Нечитайленко (2015), Документирование научного контента: современные концепции и решения, *Материалы 4-й Международной научно-практической конференции “Научное издание международного уровня – 2015: современные тенденции в мировой практике редактирования, издания и оценки научных публикаций”* р. 18–26, РАНХиГС, СПб. <http://conf.neicon.ru/materials/15-Domestic0515/Materials-0515.pdf>
- de Kemp, Arnoud (1996), Options for the Future, *Proceedings of the Joint ICSU Press/UNESCO Expert Conference on Electronic Publishing in Science* р. 149–152, ICSU Press, Paris. <http://eos.wdcb.ru/eps1/dekemp.htm>
- Mittelbach, Frank, Chris Rowley (1997), The LATEX3 Project, *TUGboat*, 18, No. 3–Proceedings of the 1997 Annual Meeting, 195–198. <https://www.tug.org/TUGboat/tb18-3/l3project.pdf>
- Nechitailenko, V. A. (2015), Record of science technologies. I. Online journal, *Geoinf. Res. Papers*, 3, BS1001, doi:10.2205/2015BS016
-