GEOINFORMATICS RESEARCH PAPERS
Proceedings of Geophysical Center RAS

ИССЛЕДОВАНИЯ ПО ГЕОИНФОРМАТИКЕ
Труды Геофизического центра РАН

VOL. 4, BS4014, doi:10.2205/2016BS041, 2016

Converting LaTeX to HTML5 and EPUB3: A case study

V. A. Nechitailenko[1]

Received 10 October 2016; accepted 1 December 2016; published 8 December 2016.

[1] The article is devoted to the development of the converter for reformatting LaTeX documents into HTML5 and EPUB3 ones. The essence of the decision proposed and implemented by the author is based on the replacement of the macros used by LaTeX class with new macros enabling to emulate the structure of XML-compliant document. A used set of macros is a finite set, that allows to build a LaTeX-to-XML/XHTML conversion algorithm. Although the proposed solution limits the ability of LaTeX it it enables a better representation of structure and content within the rules defined by the publisher for scientific publications. *KEYWORDS:* electronic book; electronic publishing; information technologies; record of science; semantic inclusion; metadescription; Crossref; eLibrary.

Citation: Nechitailenko, V. A. (2016), Converting LaTeX to HTML5 and EPUB3: A case study, *Geoinf. Res. Papers, 4,* BS4014, doi:10.2205/2016BS041.

Introduction

[2] Since the late 80-ies of the last century, the LaTeX markup language received wide distribution and has become a *de-facto* standard in the scientific community, especially in the fields of precise and technical sciences, Earth sciences and some others. The choice of LaTeX as a basic primary markup language was predetermined by the absence of real competitors, especially the lack of such for complicated mathematical and chemical texts.

[3] With the development of telecommunications and appearance of first standards of computer languages immediately arise the need for development of conversion tools of LaTeX documents to HTML and later to XML ones. Since the beginning it was clear that the correct translation of arbitrary LaTeX texts in XML is not possible, because the TeX/LaTeX is not compatible with SGML concept.

[4] That is why all known developments in the text converters of LaTeX to HTML are limited to fixed sets of processed LaTeX macros. Expanding these sets by adding the respective program modules to the converter does not eliminate the problem, although it is very helpful in terms of adopting the converter to solve specific problems (i.e. the conversion of a document with specific DTD).

[5] Very important is the understanding of a fundamental difference between the concept of XML, as an SGML subset, which defines the object's components of the document, their structure and properties, without worrying about how this document is presented at the physical layer (medium), and the TeX concept, which is essentially reduced to a description of pen-on-paper behavior only.

[6] The common view that macro packages like LaTeX, AmsTeX and some others overcome this difference is not more than misleading. In the work of well-known TeXguru *Mittelbach and Rowley* [1997, p. 196] we read:

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

[7] Referred to in this quote similarity of concepts, is not direct, but only indirect, i.e. there is no 1–1 matching.

[8] To complete the picture, let me quote another TeXguru [*Sebastian Rahtz*, 2005] to the question of a user, as to whether it possible to find a good conversion tool for LaTeX to XML:

> *... 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 :-)*

[1]Geophysical Center RAS, Moscow, Russia

Copyright 2016 by the Geophysical Center RAS.
http://ebooks.wdcb.ru/doi/2016BS041.html

From a Bad Idea to Pragmatic Solution

[9] A document (scientific article or book) includes a bit of objects that have larger or smaller semantic meaning. For better perception of these objects (different levels of headings, figures, tables, lists, including reference lists, text selection, hyperlinks and so forth) they are somehow emphasized, emphasizing rules are defined in the publisher's standards and cultural tradition, and may be the same for different objects or different for the same object class.

[10] Does this mean that for generating versions of documents, for example, in PDF/PS, and/or HTML/XML we need to have a variety of appropriately generated source files?

[11] Yes, if we want to use a full power of TEX/LATEX publishing system with its numerous extensions and the possibility of including author-defined LATEX macros directly in the source file, as well as include dynamic and interactive objects.

[12] No, if the publisher and the author are ready to use a limited list of macros, which, on the one hand, define the rules for the interpretation of the original text by TEXengine and, on the other hand, emulate XML-compatible structure, thus providing the possibility to build an algorithm of converting LATEX source file to HTML5/XHTML/EPUB3.

[13] Using XML-type languages can, in principle, not only take into account the specificities of different subject areas, but also to increase efficiency in the structuring of automated processing of articles and other documents for later indexing. However, without full ontological description of a subject field the use of XML for semantic structuring is also limited, as well as the use of LATEX which, as already was mentioned, being the presentation language contains semantics only indirectly.

[14] Most of published today electronic journals, as well as printed on paper, are human-oriented, not machine-oriented. At the same time, there is every reason to believe that in the foreseeable future scientific publications will focus primarily on the input into information intelligent systems supporting intermachine interface.

[15] And yet,.., the best solution is to follow the recommendations made at the First ICSU Press/UNESCO Expert Conference on Electronic Publishing in Science (Paris, 1996): "*Do it, then fix it!*"[2]

[16] This approach has been implemented by the author through development of ELXpaper (ELectronic eXtended paper) software package for journals published by the Geophysical Center RAS (*Russian Journal of Earth Sciences*, *Vestnik Otdelenia nauk o Zemle RAN*, and *Geoinformatics Research Papers*).

[2] See some interesting statements from the mentioned conference collected by *Arnoud de Kemp*, [1996].

ELXpaper Software Package

[17] The ELXpaper Package includes three major components: `elxpaper.sty` style file, Perl script `elxpaper.pl` with a configuration file `elxpaper.cfg`, Perl script `elxtomml.pl` with configuration file `elxtomml.cfg`, as well as a set of basic pre-built elements (CSS files, logos and layouts of covers and title pages of published books, etc.).

[18] **Style file elxpaper.sty** is an extension of the LATEX standard class `article.cls`. It supports a two-column journal format, the generation of internal and external active hyperlinks, error messages and warnings, and produces translation of a source file into DVI and PDF formats, as well as generation of XML-metadescriptions of published documents for registering them in Crossref system (in accordance with the XML-schema Crossref 4.3.3) and loading to the scientific electronic library eLIBRARY.RU (in accordance with the XML-schema eLibrary CE 7.1.4.1284).

[19] The latter problem is solved directly in the process of translation of the original text due to the fact that TEX/LATEX is not just one of the markup language of the text, and, in essence, a software system that implements the concept of Turing machine programming. Of course, programming capabilities of TEX/LATEX are significantly inferior to high-level programming languages, but they are sufficient for constructing a set of macros that extend LATEX standard classes and emulate the semantic structure of the document. The details of `elxpaper.sty` were considered in [*Nechitailenko,* 2015]

[20] **Perl script elxpaper.pl** converts a LATEX source file into a set of HTML5 files including the root file with a dynamic table of contents and set of called from the table of contents files, including figures, tables, and a number of auxiliary files. Multi-pass translation includes the following steps:

- allocation of fragments for the subsequent formation of icons (for figures and tables);
- numbering of lines of LATEX source code and merging them into paragraphs;
- splitting paragraphs into semantically significant fragments (for example, if there is a display element within a paragraph);
- processing of the preamble with extracting the basic parameters;
- allocation of the sets of protected fragments (e.g. verbatim blocks and components, inline math, etc.) for subsequent fonts substitution and replacement diacritical TEX characters with UTF-8 ones;
- extraction from the text table of contents entries, metadata and reference lists;

- generation of XML metadata for Crossref and eLibrary;
- processing of internal and external hyperlinks;
- extraction floats (figures and tables), footnotes, and tabulated fragments from LaTeX source file;
- conversion of the main document (root file) to the HTML5 format;
- conversion of the additional files (figures, tables, tables of contents, maps, etc.);
- generation of a batch file that contains instructions for translating a set of HTML files into a set of XHTML, followed by the cyclic call `elxtomml.pl` program (see below.);
- generation of XHTML file structure provided EPUB3 standard, including TOCNNN, `package.opf`, etc.;
- generation of title and cover pages for e-books in EPUB3 format based on predefined templates and specific information of a particular document (article).

[21] **Configuration file elxpaper.cfg** contains a description of all parameters processed in `elxpaper.pl` commands, including their masks (patterns), registration data of journals published by GC RAS, links to free-domain software, etc. In addition to standard LaTeX commands defined in `article.cls` (some are overridden, e.g., title, abstract, etc.), also included are series of commands, in certain packages extending `article.cls`, in primarily `hyperref.sty` package. The list of used LaTeX packages is included in the preamble of `elxpaper.sty`.

[22] The `elxpaper.pl` script operates under the assumption that (i) the source file of an article in LaTeX does not generate error messages when it is translated by `latex` or `pdflatex` drivers, (ii) `jobname.aux` file presents in the current directory, (iii) there are no used-defined macrodefinitions in the preamble and the text of the article, generated by `\newcommand`, `\renewcommand`, `\def`, `\gdef`, etc.

[23] All commands in the text, which are used only for text formatting and have no semantic meaning, such as `\space`, `\skip`, `\linebreak`, etc. are ignored, as well as commands with semantic meaning, but for which the processing algorithm is not defined. Commands for the font management are replaced by the corresponding definitions in the `elx-online.css` CSS file.

[24] **Perl script elxtomml.pl** converts HTML5 files, generated in the previous step, into XHTML files necessary for inclusion of the latter into EPUB3 structure, wherein some elements of the original HTML5 file structures are replaced by more stringent XHTML ones. At that stage Russian-language documents are performed to Unicode.

[25] Script `elxtomml.pl` also solves the problem of converting mathematical and chemical formulas (both display and in-text) initially written in LaTeX notifications to MathML format. As one knows, the publishing system TeX/LaTeX/AmsTeX provides a sort of *de facto* standard to represent math for documenting scientific content. With the transition from the POP (printed-on-paper) publications to electronic publications there was a need to find a way to present TeX mathematics into a form that could be reflected in the Web browser window. The first implementations of such solutions are widely used images for replacement of mathematical and chemical formulas. Although this decision could be implemented using fairly simple means, it had a number of serious deficiencies, the main of which was the problem of scaling the image and the problem of interpretation of such images by search engines.

[26] With the development of the *MathMl* markup language, which became a part of the HTML5 standard, officially recommended by the W3C Consortium and registered in 2015 as a standard of the International Organization of Standards (ISO/IEC DIS 40314), the problem is reduced to the conversion of mathematics, presented in TeX/AmsTeX into MathMl format. Currently, among the many solutions of this problem a special place is taken by the *MathJax* system developed and maintained by the MathJax consortium – a joint venture of the American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM).

[27] MathJax is not a traditional offline file converter. The conversion is performed "on the fly" when loading an HTML file to the browser, which has in its body mathematical fragments in TeX notations, and in the preamble (<head>) command calling MathJax scripts either from CDN network servers (https://www.mathjax.org/), or from a separate copy of MathJax, installed on the user's server. MathJax uses CSS with web fonts or SVG, instead of pictures (bitmap or Flash), so math is scaled along with the text at all levels of zoom in/out. This conversion results are sent directly to the screen.

[28] Conversion of mathematical texts for HTML5 and EPUB3 is implemented in the ELXpaper package in two different ways. For the online version of the article in HTML5 format it is enough to save the math in the form as presented in the original LaTeX file and add the scripts calling MathJax into the preamble (<head>).

[29] Note that in the version of the ELXpaper package, developed and presented here, MathJax scripts are called directly from `cdn.mathjax.org` network sites, so user's computer must be connected to the Internet. You can also install MathJax on a local server (see details). The second solution in some cases looks more preferable, but it will require regular updating of the local version of MathJax, while direct connection to the MathJax CDN eliminates this.

[30] For e-books in EPUB3 format this solution is not suitable, as an e-book in accordance with the recommendations of the IDPF (International Digital Publishing Forum) should ensure the correct interpretation of the texts in the absence of access to the network (offline). EPUB3-compliant reading devices as well as emulators of such devices on desktops, in accordance with the recommendations of the IDPF should

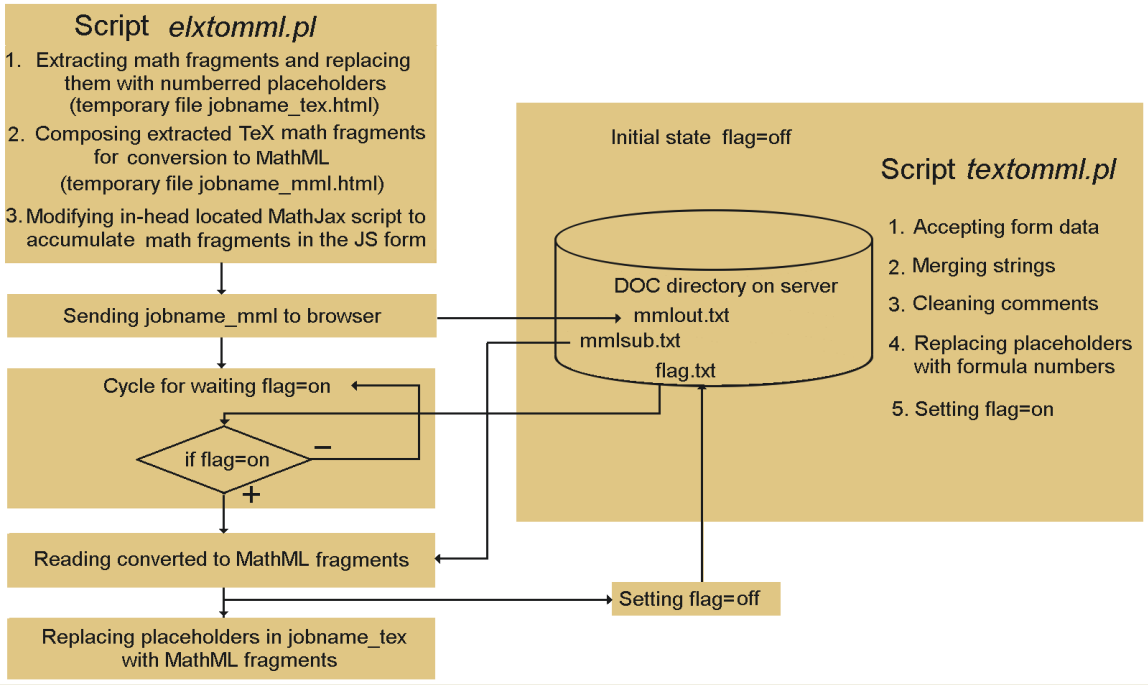

Figure 1. Flow-chart of interaction `elxtomml.pl` and `textomml.pl` in the working directory and the `doc` directory of the ancillary server.

provide the correct HTML5/XHTML interpretation, including MathML.

[31] As is known, the EPUB format is a compressed archive of all files that make up publication and meet the XHTML standard, with the addition of files that define the file structure (`package.opf`) and type (mimetype). So, when transferring HTML5 files to a more strict XHTML format, we need to convert mathematical TEX fragments (in-text and displayed) into fragments of MathML.

[32] The analysis of existing LATEX-to-MathML converters showed that in many respects they can not compete with the MathJax system (due to restrictions on the source files, lack of support, the commercial terms of use, etc.) At the same time, though MathJax creates MathML fragments for each of TEX fragments with an option to view them via pop-up windows, one cannot output them into a separate file due to the JavaScript inherent limitations.

[33] We have found and implemented a solution that TEX creator Donald Knuth [*Knuth,* 1984] could attributed to the "dirty tricks" category. Its essence is that from JavaScript, we can not write down the generated MathML file on a local computer, but can send it as a content of HTML form to a server containing the program for processing form content with recording result to an external memory. The simplest way to do this is to install a dedicated web server directly onto the localhost, where the ELXpaper package installed. Then the result of the server processing will be obviously available to appropriate package program. The synchronization of these programs can be completely solved through an ordinary system of flags. If for some reasons the installation of such server on localhost is impossible or undesirable, for example when different computers are used for preparation of articles for publication, the corresponding scripts directory for recording the converted files can be installed onto any available web server.

[34] **Configuration file elxtomml.cfg** determines the order of communication of `elxtomml.pl` with auxiliary programs (Figure 1) interaction with the MathJax system, including address directories `doc` and `cgi-bin` of the dedicated web server, the type of the browser used in the conversion process and commands of its opening and closing, service files addresses of generated EPUB3 archive.

[35] **Additional package components** are the set of basic pre-built elements (logo images and templates of the cover and title pages, CSS files, diacritical LATEX symbols matching UNICODE characters, etc.) See Appendix for more detail.

Conclusion

[36] The developed technology is being used successfully in practice of editorial preparation of publications of the Geophysical Center RAS, the first of all by Russian Journal of Earth Sciences. Each of published in RJES articles, since June 2012, is presented in four formats: (i) the basic version (version of record) in PDF format, which includes internal and external hyperlinks as well as dynamic and interactive

content; (ii) the version in HTML5 format; (iii) the version in EPUB3 format; and (iv) the version in PDF format adapted for small screens (e-readers and smartphones).

[37] One should keep in mind that at the time of writing this article most of the devices and programs for reading EPUB3 documents does not provide a perfect playback. Our tests have shown that the best results are provided by iBooks2 application on the iPad and iPhone, Readium, recommended by IDPF as a basis for inclusion in software for tablets, smartphones and e-books.

[38] All the generated versions in the EPUB3 format are checked for conformance with the EPUB3 standard using `epubcheck3.0` program and EPUB Validator, recommended by the IDPF, and then uploaded to the server if no error messages are shown.

[39] **Acknowledgment.** Author is grateful to his colleagues at Eastlake Crossing Group for their advices and testing presented software.

Appendix

[40] The updated and translated edition of the package is available as an archive elxpaper_1.5_en.zip on the GC RAS server dedicated for electronic publishing. The archive file contains all necessary information for installing and using of the package. The archive also includes the original version of this article in LaTeX format, which was used to produce its PDF, HTML5, and EPUB3 versions.

References

de Kemp, Arnoud (1996), Options for the Future, *Proceedings of the Joint ICSU Press/UNESCO Expert Conference on Electronic Publishing in Science* p. 149–152, ICSU Press, Paris. (http://eos.wdcb.ru/eps1/dekemp.htm)

Knuth, Donald E. (1984), *The TeXbook*, 483 pp., Addison-Wesley, New York.

Mittelbach, Frank, Chris Rowley (1997), , 1999 The LaTeX2$_\varepsilon$3 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

V. A. Nechitailenko, Geophysical Center RAS, Moscow, Russia (vitaly@wdcb.ru)