RESEARCH ARTICLE | JANUARY 05 2017

Web portal for dynamic creation and publication of teaching materials in multiple formats from a single source representation FREE

E. A. Roganov; N. A. Roganova; A. I. Aleksandrov; A. V. Ukolova

AIP Conf. Proc. 1797, 020016 (2017)
https://doi.org/10.1063/1.4972436

Web Portal for Dynamic Creation and Publication of Teaching Materials in Multiple Formats from a Single Source Representation

E. A. Roganov[1, a)], N. A. Roganova[1, b)], A. I. Aleksandrov[1, c)], A.V. Ukolova[1, d)]

[1]*National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe shosse 31, Moscow, 115409, Russia*

[a)]Corresponding author: earoganov@mephi.ru
[b)] naroganova@mephi.ru
[c)] aialeksandrov@mephi.ru
[d)] avukolova@mephi.ru

Abstract. We implement a web portal which dynamically creates documents in more than 30 different formats including html, pdf and docx from a single original material source. It is obtained by using a number of free software such as Markdown (markup language), Pandoc (document converter), MathJax (library to display mathematical notation in web browsers), framework Ruby on Rails. The portal enables the creation of documents with a high quality visualization of mathematical formulas, is compatible with a mobile device and allows one to search documents by text or formula fragments. Moreover, it gives professors the ability to develop the latest technology educational materials, without qualified technicians' assistance, thus improving the quality of the whole educational process.

INTRODUCTION

Web technologies have become an important part of the toolkit, used in any form of education. Nowadays every modern university has a web portal, where a student can find tests and necessary information needed in his study. The portal structures and the kinds of tests and information can be different. Learning management system Moodle [1], for example, provides more than ten file formats.

Faculty members often create training materials on a home desktop using one of the office technology formats (e.g. using Microsoft Office or LibreOffice). With the help of LaTeX document markup language professors can create pdf-documents with mathematical formulas, e.g. MIT educational materials [2]. Some faculty members are able to use the newest web technologies and design easy-to-use hypertext documents and presentations. It has become already popular [3].

As a result such web portals often contain information, which is unstructured and doesn't meets modern requirements. It's difficult to enter and use them from mobile devices and the search documents by text or formula fragments is usually impossible.

Faculty members who use the newest technologies for creating materials face significant technical difficulties installing and keeping up-to-date required software on their desktops.

That's why it is important to create a web-portal that will help university professors to compose training materials using the newest technologies without qualified technicians assistance.

WEB-ORIENTED TRAINING MATERIALS

Nowadays mobile devices are widely used that's why we need training materials, work with which on such devices is a convenient and comfortable. Viewing PDF documents and files in the Word or Excel formats on such

Information Technologies in Education of the XXI Century (ITE-XXI)
AIP Conf. Proc. 1797, 020016-1–020016-7; doi: 10.1063/1.4972436
Published by AIP Publishing. 978-0-7354-1463-1/$30.00

devices is complicated. Therefore, actually, it is equivalent to the requirement of preparation of materials in an html-format. Materials for students can be represented as web page or hypertext presentation [4]. For example, "two-dimensional" Reveal.js presentation [5] may be considered as alternative traditional presentations prepared using PowerPoint or Beamer.

Visualization of mathematical formulas in browsers was a problem for a long time. This was a reason for rarely used web-oriented education material. All solutions offered before the creation of JavaScript-library MathJax [6], had certain disadvantages. MathJax allows to include a source code of mathematical formulas which are written down with use of MathML, LaTeX or ASCIIMathML in the html-document. While viewing the page in the browser the interpretation of this code takes place and the user can see a beautiful formula.

Many physicists, chemists, mathematicians and programmers familiar with computer typesetting system LaTeX, widely used in the scientific world for the production of publications. Especially useful is the ability to turn on formulas in TeX-notation [7] – one of the best ways to present mathematics. Figure 1 shows a fragment of html-page containing formulas and a view this page in your browser.

Learning to use TeX-notation to enter formulas is easy, as is very comfortable and natural. Many famous websites (for example, Wikipedia) for placement on the pages of mathematical texts use for a long time the graphics received by means of this technology. MathJax eliminates the prior graphic phase images creation and allows to include a TeX-notation formula directly to the web page.

```
<p>
 Root <span class="math" >\(x=-3\)</span> of equation
 <span class="math" >\(x^{3}-7x+6=0\)</span>
  with the formula Gerolamo Cardano is expressed as
</p>

<p>
 <span class="math">
   \[\sqrt[3]{-3 + \frac{10}{9}\sqrt{-3}}+ \sqrt[3]{-3 - \frac{10}{9}\sqrt{-3}}.\]
 </span>
</p>
```

Root $x = -3$ of equation $x^3 - 7x + 6 = 0$ with the formula Gerolamo Cardano is expressed as

$$\sqrt[3]{-3 + \frac{10}{9}\sqrt{-3}} + \sqrt[3]{-3 - \frac{10}{9}\sqrt{-3}}.$$

FIGURE 1. TeX-notation formula in HTML document.

MARKDOWN & PANDOC

So, we need to prepare materials in html-format. At the same time we want to have the such materials in the form of high-quality pdf-documents and Microsoft-compatible files. It would also be nice to get the materials in e-book-compatible and in some other formats. Is it possible?

To implement this approach we need a convenient prime language to record the original text. From this initial form we can get a range of beautifully designed documents in various formats: hypertext materials, printable pdf-materials, hypertext pdf-presentations, docx-papers, etc.

The markup language Markdown [8] is good for such work. It was originally designed as a convenient language for reading and directly editing from which it is easy to get html. In a series of articles (see e.g. [9, 10]) one can find specific recommendations how to use this language for preparation of scientific publications and materials. Markdown is so simple that everyone can learn how to use it in only 5-10 minutes. For example, a piece of text source fragment from which the html-code and formula shown on Fig. 1 is illustrated on Fig. 2.

The creation of document converter Pandoc [11] (a library, providing high-quality conversion of documents from a number of input formats to a wide variety of output formats) confirmed the role of Markdown as a

"universal" language for internal representation. From a text in Markdown language Pandoc can produce more than three dozen of documents in different formats.

```
By using Cardan's formula, the root $x=-3$ of the equation
$x^{3}-7x+6=0$ may be expressed as

$$\sqrt[3]{-3 + \frac{10}{9}\sqrt{-3}}
+ \sqrt[3]{-3 - \frac{10}{9}\sqrt{-3}}.$$
```

FIGURE 2. Source code.

Since Markdown itself is a very simple language with limited visual possibilities, numerous extensions of the language have already been created, in particular, GitHub-Flavored Markdown and PHP Markdown Extra. Figure 3 illustrates Pandoc capabilities which are especially important for our purposes.

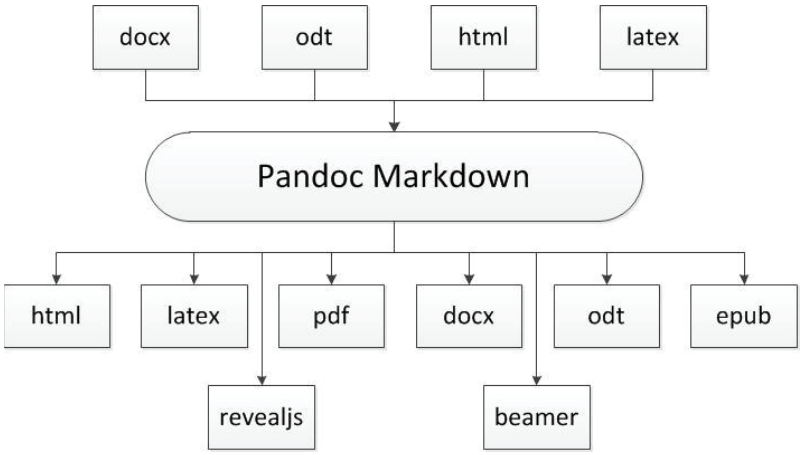

FIGURE 3. Pandoc capabilities.

Using Markdown & Pandoc together in order to obtain a number of output documents in various formats from a single source representation can already be considered as a standard approach, although this has not been a widespread yet. A description of all main ideas of this approach is published, for example, in [12].

A detailed description and some recommendations and improvements for document converter Pandoc (beyond the scope of our work) can be found in paper [13].

NECESSARY SOFTWARE AND EXPANSION OF THE MARKDOWN LANGUAGE

Pandoc is a very effective tool, but using only this document converter is not optimal for processing of markdown material. In order to make the process of producing output documents in various formats more convenient it is reasonable to use additional software. Utility Make, which is well-known among programmers, is one on the simplest example of such software. For reasons that will become clear later, it is more convenient for us to use its Ruby-analogue, i.e. Rake [14].

In order to create pdf-documents LaTeX should be installed to the system. Moreover, we need to have latexmk to allow modification of tex-documents obtained from a markdown source. MathJax and Reveal are libraries which provide visualization of mathematical formulas in a browser and the ability to create hypertext presentations, respectively.

Besides text and mathematical formulas it is often necessary to include images and fragments of programs in a particular programming language. In order to make a full use of images in svg format, one should install vector graphics editor Inkscape.

As mentioned above, Markdown has a few means of expression and therefore one of its extensions is often used in practice. However, even these extensions sometimes do not provide necessary features.

For example, our task is to prepare an initial text, from which we should obtained a hypertext (html) or printed (pdf) material as well as presentations for lectures. The problem is not all text should be shown in our presentation. Furthermore, Pandoc's marks for separating one presentation slide from the other will be visible as horizontal lines in the output document.

There are some more reasons why is it better to use an expansion of Markdown language and preprocessor, which will convert the source code to "standard" Markdown, later processed by Pandoc. It is better to use utility m4 [15], the traditional UNIX macro processor.

Figure 4 shows a set of languages, software and technologies, used in preparation of output documents from Markdown language. All these tools are free, many of them are included in standard Linux-distributions, and other can be easily installed.

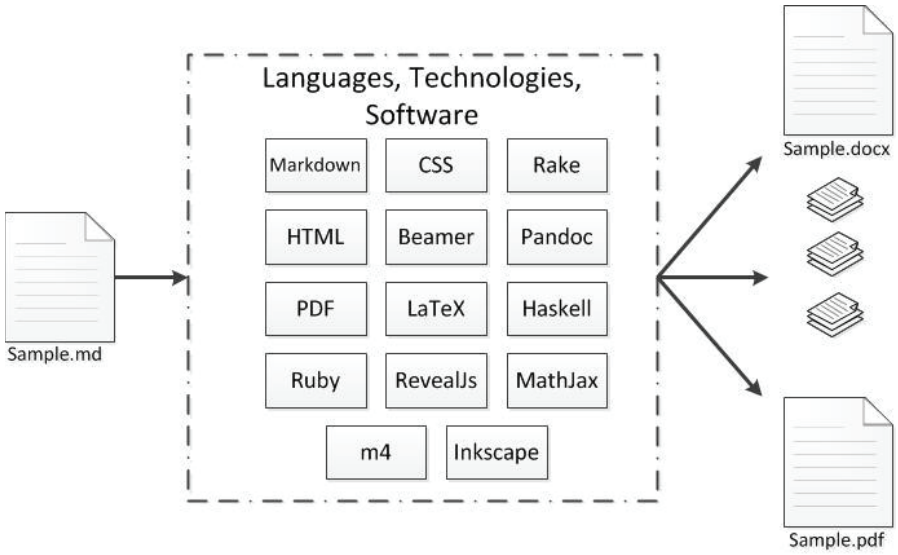

FIGURE 4. Software and technologies.

There is, however, the following question: can a typical professor of physics, chemistry or mathematics cope with such task? And does he want such difficulties? Probably not. It is easier for him to continue using of Microsoft Office or LibreOffice. At best, he will create pdf-documents using LaTeX.

We can "persuade" professors to use the latest technology only by hiding from them all its complexity. Of course, they need to learn Markdown and how to use TeX-notation for formulas, but everything else should be invisible for them. At the same time they do not have to install any software on their computer. They generally should be able to use mobile devices.

The development of a special web portal can ensure them to do it. According to Ruby on Rails [16], we are able to implement a web portal which dynamically creates training materials in various formats from uniform initial representation according to the chosen templates and a set of parameters.

We have significantly expanded the list of software tools and technologies shown on Fig. 4. And it will be convenient for faculty members.

THE EDUCATIONAL PROCESS SUPPORT PORTAL

Ruby on Rails is a web framework widely used for rapid development educational web portals (e.g. [17]). The framework is an ideal instrument for solving the problem of dynamic creation and publishing educational material in multiple formats from a single source.

The main models are:

- source files (in Markdown format);
- additional files – graphic images, style sheets (applied to hypertext documents), source code files;

- documents (generated by Pandoc).

A faculty member can create a new document on the portal directly from the browser using Ace web editor [18] or upload a document from his or her computer. One can also upload documents in docx, odt, epub, docbook, html, latex and some other formats.

Additional files can be uploaded to the portal separately or as a zip-archive.

For each of materials it is possible to set the required formats of output documents with the indication of padding parameters. These parameters modify both contents, and appearance of the document. For a pdf-file such parameters can be, for example, font-size in a latex-file or pagination requirements. By means of parameters it is also possible to include in the output document particular padding fragments of a starting material. For example, answers to tasks which are necessary only in version of the output document for teachers. Other example is the hints used only in slides to lectures to the sorted tasks.

Figure 5 shows a workflow diagram of the educational portal from the user's point of view.

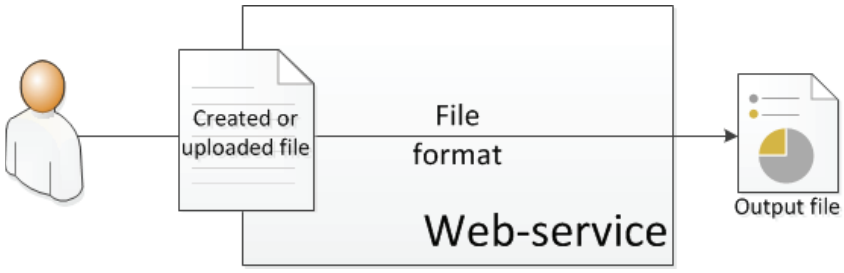

FIGURE 5. Workflow diagram of the educational portal.

When output parameters of the document are set, the program that allows to obtain specified document starts. It converts source material into output document using Pandoc. In many cases pre- and post- task execution is required. Figure 6 shows the general workflow of converting a source material into an output pdf-document.

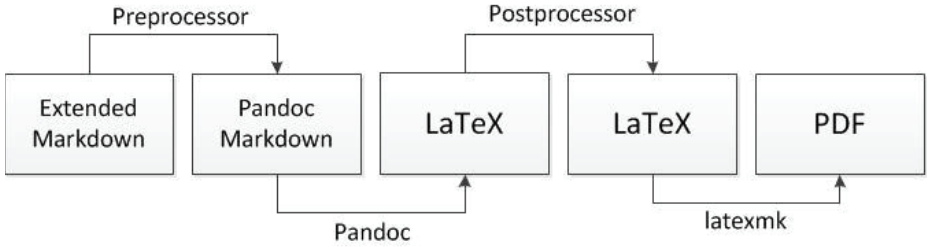

FIGURE 6. General workflow of converting a source material.

SOME IMPLEMENTATION DETAILS

The educational process support portal can be used for:

1. creating outcome documents and publishing them on the other (educational) portal;
2. students' direct access to content.

In the first case the portal's users are only professors. There is an additional problem in this case: converter Pandoc can not create html-documents that contain both MathJax math and embedded graphics.

A simple way to solve this problem is packaging html-document and related graphic files into zip-archive. In that case the educational portal will perform the correct visualization of the uploaded archive.

Regardless of usage of the educational process support portal it is necessary to allow supporting svg-files in the source materials. There's a problem that LaTeX does not support includegraphics directive of svg-format files as a parameter. Inkscape vector graphics editor should be installed on a server to solve this problem.

The educational process support portal allows to keep all training materials in uniform standard formats. Minor modifications of standard templates of Pandoc converter allow pasting the university logo and other identifying information into documents. We also have a possibility to include a number of version and the date of last modification. Figure 7 illustrates the realization of this possibility.

Text or formula search can be easily implemented in Markdown that includes TeX-notation formulas. For this purpose Elasticsearch [19] engine or PostgreSQL database [20] can be used. In the latter case usage of gem PgSearch [21] is recommended.

```
rule /\.(html|tex)$/, [:date, :version] =>
  [proc do |file|
     file =~ /(.*?)(_.*?)*\.(html|tex)/
     ["#{$1}.md"] + FileList["#{LIB}/*"] + FileList['files/css/*'] +
       FileList['files/programms/*'] +
       ($3 == 'tex' ?  FileList['files/images/*'] : [])
   end].flatten do |t, args|
     ...
end
```

FIGURE 7. Rakefile fragment.

Html-documents with math formulas created by Pandoc converter are incorrect in the following way: non-breaking space in source text before math formula does not give the desired effect. A similar problem occurs when punctuation marks follow a formula immediately.

To solve these and other problems a simple postprocessor is required. It has to process the html-file which comes from converter Pandoc. For the material in Russian there is additional problem. At creation of PDF versions of the Russian-language documents containing accent symbols padding transformation of TEX-files is necessary. Figure 8 shows a fragment of a corresponding filter.

```
hash = {
  'á' => "\\\\'{a}", 'é' => "\\\\'{e}", 'и́' => "\\\\'{и}",
  'ó' => "\\\\'{o}", 'ý' => "\\\\'{y}", 'ы́' => "\\\\'{ы}",
  'э́' => "\\\\'{э}", 'ю́' => "\\\\'{ю}", 'я́' => "\\\\'{я}"
}
puts buf.gsub(/á|é|и́|ó|ý|ы́|э́|ю́|я́/, hash)
```

FIGURE 8. Characters filter.

It can be useful to have an option that disallows students “to look at the next slide” (containing a hint or an answer to a question) during lectures or practical classes. The problem can be solved by creating an additional web portal and uploading ordinary revealjs-presentations to it. While uploading the presentations should be processed and transformed into a sequence of slides. The access to them should be set by the professor.

CONCLUSION

According to ideas mentioned in this paper, we implement a portal which is convenient for practical use by teachers. Even for creation of the complex methodical materials. The use of this portal requires neither deep knowledge of latest of information technology, nor the installation of special software. The portal is compatible with mobile devices and any qualified faculty member can easily get a bit of additional knowledge and skills necessary to work with it.

The portal contains only free software and provides:

- dynamic producing from a single original material source of documents in more than 30 different formats, including html, pdf and docx and various presentations;

- convenient TeX-notation for creating mathematical formulas and for visualizing them in a high quality in most of common formats of output documents;
- flexible design of html-output documents facilitating the work via mobile devices;
- the ability to search documents by text or formula fragments;
- creation of presentations not allowing students "to look at the next slide" (containing a hint or an answer to a question) during lectures or practical classes;
- the development of the latest technology educational materials by faculty members without qualified technicians assistance;
- improvement of the quality of the whole educational process.

REFERENCES

1. Moodle – learning management system, available at https://moodle.org.
2. MIT OpenCourseWare – educational materials of the Massachusetts Institute of Technology, available at http://ocw.mit.edu.
3. P. Sefton, Serials Review **35**(3), 154–158 (2009).
4. Web-based slideshow, available at https://en.wikipedia.org/wiki/Web-based_slideshow.
5. H. El Hattab, Reveal.js – The HTML Presentation Framework, available at http://lab.hakim.se/reveal-js.
6. A JavaScript display engine for mathematics, available at http://www.mathjax.org.
7. Typesetting system TeX, available at https://en.wikipedia.org/wiki/TeX.
8. J. Gruber, Markdown: A text-to-HTML conversion tool for Web writers, available at http://daringfireball.net/projects/markdown.
9. J. Voegler, J. Bornschein, G. Weber, "Markdown – A simple syntax for transcription of accessible study materials," in *Proceedings of the 14th International Conference on Computers Helping People with Special Needs (ICCHP),* edited by K. Miesenberger, D. Fels, D. Archambault, et al., Paris, 2014 (Springer, Information Systems and Applications…, 2014), Vol. 8547, pp. 545–548.
10. S. Ovadia, Behavioral & Social Sciences Librarian **33**(2), 120–124 (2014).
11. Pandoc Document Converter, available at: http://pandoc.org.
12. A. Kielhorn, TUGboat **32**(3), 272–277 (2011)
13. J. Krijnen, D. Swierstra, and M. O. Viera, "Expand: Towards an extensible Pandoc system," in *Proceedings of the 16th International Symposium Practical Aspects of Declarative Languages (PADL),* San Diego, 2014 (Springer, Lecture Notes in Computer Science, 2014), Vol. 8324, pp. 200–215.
14. A make-like build utility for Ruby, available at https://github.com/ruby/rake.
15. Traditional Unix macro processor, available at http://www.gnu.org/software/m4/m4.html.
16. Ruby on Rails: Web Development That Doesn't Hurt, available at http://rubyonrails.org.
17. D. T. Irribarra, R. Freund, W. Fisher, M. Wilson, Journal of Physics: Conference Series **588**(1), 012042 (2015).
18. High performance code editor for the web, available at https://ace.c9.io.
19. S. Banon, Elasticsearch: An open source, distributed, RESTful search engine, available at http://www.elasticsearch.org.
20. Object-relational database management system PostgreSQL, available at http://www.postgresql.org.
21. PgSearch – Ruby gem to take advantage of PostgreSQL's full text search, available at https://github.com/Casecommons/pg_search.