What is this?

Rodrigo Hernández Mota

What is this?

The Short Answer

Q: What is this?

A: A site where I host and publish the documentation and reference materials for my talks.

The Long Answer

Motivation

Have you ever feel limited by the "expressiveness" of the documentation tools you use (e.g., simple markdown)? Or frustrated by the complexity they might introduce (e.g., latex)? Well, this situation inspired me to enter a quest to find the best technical writing tool for (software) engineers that's simple enough to learn, has a lightweight syntax, and avoids unnecessary complexity and boilerplate. I recently concluded that such a tool doesn't exist (but could!).

Since my background is on machine learning and data engineering, I was looking for something that can parse mathematical equations and at the same time, execute arbitrary code snippets. In particular, I was looking for:

- **A expressive but straightforward syntax for formatting.** We all know that LaTeX is the king for document formatting, but introduces significant boilerplate for small projects. On the other hand, markdown is very simple but compromises expressiveness.
- **Ability to parse and present math equations.** This is the main reason I keep coming back to LaTeX. Is there a way out?
- **Ability to execute arbitrary code snippets in several languages.** I don't ask for much. As an engineer specialized on the data fields, I expect to be able to run at least Scala, Python, and Bash.
- **Being able to use a single-source document to generate multiple outputs formats.** I might want to share the same document as a PDF, or a webpage.
- **Reference management simplification.** Nice to have!

Fortunately, there are plenty of open source tools that aim to solve particular doc-oriented problems. My solution turned out to be a **giant hack** that combines several of these tools.

The proposed approach

You might be reading this document on a website, pdf-file, or even an ebook. A single source file generated all of these outputs. Let's be clear; this is not magic. It's the result of using several high-quality open source projects. In particular, I want to express my appreciation to all the collaborators of the following projects:

- Pandoc - a Haskell based universal document converter.
- Rmarkdown - a tool for reproducible research that allows computing code and narrative to be in the same document.
- Bookdown - an Rmarkdown extension.
- Scala Tut - an SBT plugin that allows executable scala-code snippets in Markdown.

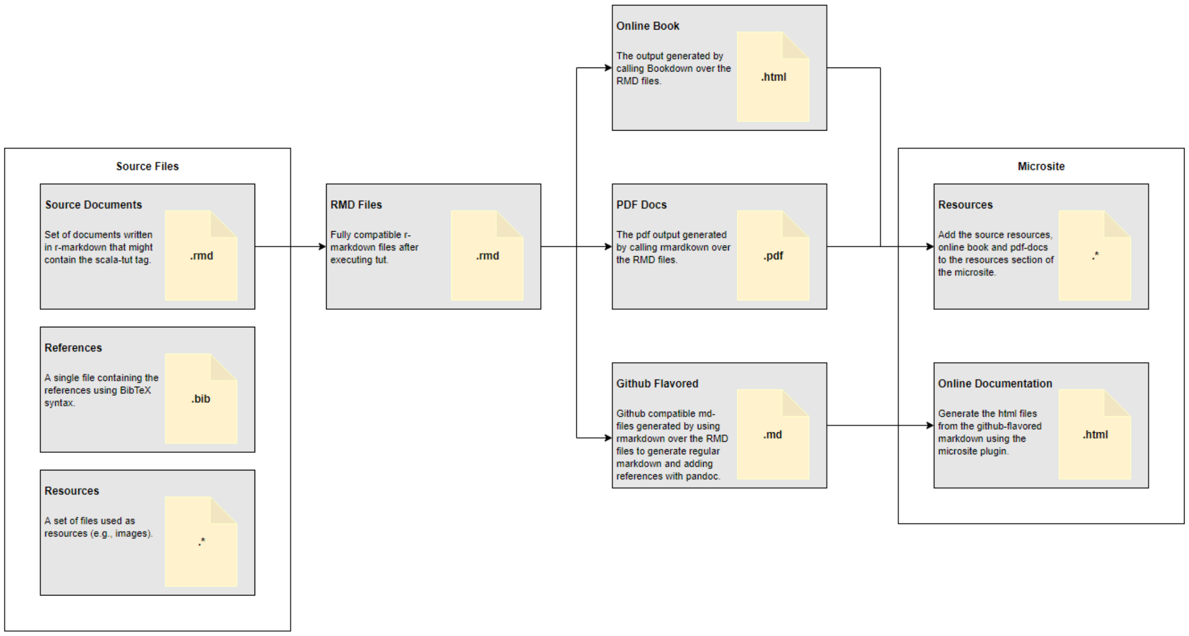
Source Files
Source Documents
Set of documents written in r-markdown that might contain the scala-tut tag.
.rmd
References
A single file containing the references using BibTeX syntax.
.bib
Resources
A set of files used as resources (e.g., images).
.*
RMD Files
Fully compatible r-markdown files after executing tut.
.rmd
Online Book
The output generated by calling Bookdown over the RMD files.
.html
PDF Docs
The pdf output generated by calling rmardkown over the RMD files.
.pdf
Github Flavored
Github compatible md-files generated by using rmarkdown over the RMD files to generate regular markdown and adding references with pandoc.
.md
Microsite
Resources
Add the source resources, online book and pdf-docs to the resources section of the microsite.
.*
Online Documentation
Generate the html files from the github-flavored markdown using the microsite plugin.
.html

Figure 1: Documentation Process

- Scala Microsite - an SBT plugin that allows the creation of microsites.

Let me explain how to combine all these tools into a giant hack:

Find a naive implementation of this process in the github repo that contains the source code for this site.

Why R markdown?

As you might have noticed, this Frankenstein tool relies mostly on R-markdown for the most relevant features. Let me cite the author of the Bookdown framework to explain my decision:

> “R Markdown may not be the right format for you if you find these elements not enough for your writing: paragraphs, (section) headers, block quotations, code blocks, (numbered and unnumbered) lists, horizontal rules, tables, inline formatting (emphasis, strikeout, superscripts, subscripts, verbatim, and small caps text), LaTeX math expressions, equations, links, images, footnotes, citations, theorems, proofs, and examples.”

(Xie, Allaire, and Grolemund 2018)

Publishing

The SBT microsite plugin facilitates the deployment of the resulting site into github pages. This service can host and serve static sites without a problem by using Jekyll.

I use a custom script that execute the whole pipeline: * `./publish --local`: serve the site in localhost. * `./publush --site`: serve the site in github pages.

Note that you must run the `setup` scripts before.

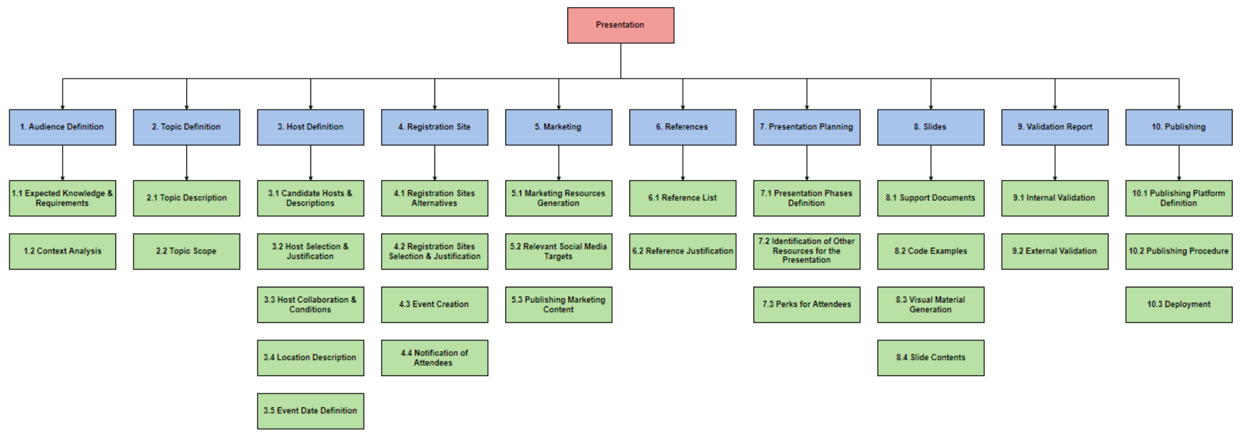
Presentation
1. Audience Definition
1.1 Expected Knowledge & Requirements
1.2 Context Analysis
2. Topic Definition
2.1 Topic Description
2.2 Topic Scope
3. Host Definition
3.1 Candidate Hosts & Descriptions
3.2 Host Selection & Justification
3.3 Host Collaboration & Conditions
3.4 Location Description
3.5 Event Date Definition
4. Registration Site
4.1 Registration Sites Alternatives
4.2 Registration Sites Selection & Justification
4.3 Event Creation
4.4 Notification of Attendees
5. Marketing
5.1 Marketing Resources Generation
5.2 Relevant Social Media Targets
5.3 Publishing Marketing Content
6. References
6.1 Reference List
6.2 Reference Justification
7. Presentation Planning
7.1 Presentation Phases Definition
7.2 Identification of Other Resources for the Presentation
7.3 Perks for Attendees
8. Slides
8.1 Support Documents
8.2 Code Examples
8.3 Visual Material Generation
8.4 Slide Contents
9. Validation Report
9.1 Internal Validation
9.2 External Validation
10. Publishing
10.1 Publishing Platform Definition
10.2 Publishing Procedure
10.3 Deployment

Figure 2: Presentation Work Breakdown Structure

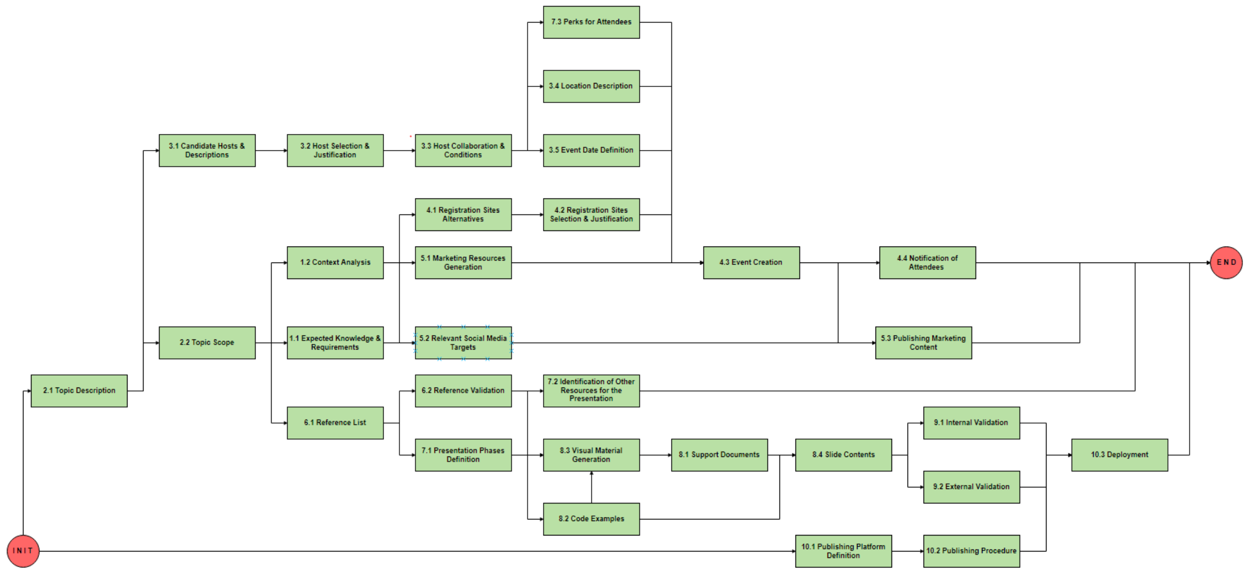
7.3 Perks for Attendees
3.4 Location Description
3.1 Candidate Hosts & Descriptions
3.2 Host Selection & Justification
3.3 Host Collaboration & Conditions
3.5 Event Date Definition
4.1 Registration Sites Alternatives
4.2 Registration Sites Selection & Justification
1.2 Context Analysis
5.1 Marketing Resources Generation
4.3 Event Creation
4.4 Notification of Attendees
END
2.2 Topic Scope
1.1 Expected Knowledge & Requirements
5.2 Relevant Social Media Targets
5.3 Publishing Marketing Content
2.1 Topic Description
6.2 Reference Validation
7.2 Identification of Other Resources for the Presentation
6.1 Reference List
9.1 Internal Validation
7.1 Presentation Phases Definition
8.3 Visual Material Generation
8.1 Support Documents
8.4 Slide Contents
10.3 Deployment
9.2 External Validation
8.2 Code Examples
INIT
10.1 Publishing Platform Definition
10.2 Publishing Procedure

Figure 3: Presentation - Work Package Network Diagram

Presentations

We can represent the project of generating a presentation with a work-breakdown structure diagram (WBS). To represent the dependencies, we can take the "leaves of the tree" and arrange them in a network diagram.

References

Xie, Yihui, JJ Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. CRC Press.