Introduction to Plain Text Workflows and Sustainable Publishing

Till Grallert

2017-02-22 17:13:32 +0100

*DRAFT*

Disclaimer 1: many of the ideas have been inspired by the work of Alex Gil (@[elotroalex](https://twitter.com/elotroalex)) and Denis Tenen (@[dennistenen](https://twitter.com//dennistenen))[[1]](#footnote-1) and discussions with the two of them and others at DHI Beirut, THATCamp Beirut, and DHSI.

Disclaimer 2: I am teaching a short workshop on some of the ideas in this post at [Digital Humanities Institute - Beirut](https://dhibeirut.wordpress.com/) on Friday, 10 March 2017. Basic slides will be made available [here](https://tillgrallert.github.io/Slides/2017-dhi-beirut/).

# Intro

In the world of (academic) publishing, large aggregators and indexers {>>could be named here<<} have turned into and acquired publishing presses and generate obscene profits by charging the public (every tax-payer worldwide) multiple times over. First by charging the predominantly publicly-funded academic for publishing the results of her publicly-funded research and by enforcing a culture of *pro bono* labour among academic reviewers and editors; second by selling this content to equally predominantly publicly-funded libraries, which then increasingly demand access fees from members of the public, who want to access their collections; and third by offloading the cost of long-term preservation to, again, publicly-funded institutions. This system not only created a hierarchy of academics and institutions in the relatively well-off “West”—two classes divided by their ability to pay for being published and accessing publications (their own and others). It also increasingly prevents anybody outside western academia from accessing cutting-edge research and participating in intellectual discourse.

One of the opportunities afforded by the *digital humanities* and the stated goal of this {==course==}{>>workshop<<} is to remove the middlemen—be they technical or entrepreneurial—between authors, readers, and the library-cum-archive. There are two main obstacles to this aim:

1. copyright laws and
2. the alienation of us, authors and academics, from the means of production.

We will not be able to change copyright legislation and the vested business interests in sustaining and expanding the regime of profit-generating copy and distribution rights in the foreseeable future, but we can all provide our knowledge under a [creative commons licence](https://creativecommons.org/). In order to do so, we need to re-claim the means of production. We argue that by doing so the main argument for restrictive copyright—namely, the provision of allegedly expensive services such as quality control and metadata curation—collapses. By the time of writing, printing costs and the global distribution of heavy and voluminous books are already negligible as the main avenue of scholarly publication, the journal, has already moved to digital online publication.

# Principles

The main principles in our effort to (re)claim the means of production are: accessibility, simplicity, sustainability, and credibility. They shall pertain both to the intellectual endeavour and to the tools employed.

1. *Accessibility*. Accessible means first of all free and open to use and re-purpose for all that have the technical ability to do. Therefore we will have to forfeit all proprietary software and formats. From the imperative of openness and accessibility derive the second and third principles:
2. *Simplicity*: apart from the necessity to be able read and write in a specific human language, there should be as few tools, hardware and software requirements as possible. Ideally everything should work on ten-year old hardware and nothing but the software packaged with the operating system—or even better: the core technologies should work without a computer.
3. *Sustainability*: only simple systems adhering to widely accepted standards can be sustained with minimal / reasonable effort.
4. *Credibility*: Credibility is at the core of scholarly production. In addition to transparency as to the sources, methodology, and tools used, authorship needs to be ascertained and acknowledged as the main tool of scholarly quality control.

# Tools / formats

## 1. Writing

Looking at the most broadly-employed software in academic contexts and beyond, Microsoft’s Word, a piece of bloated and expensive proprietary software, what are the functions we need to replace?

### 1. composition / authoring

To avoid overtly complex software and proprietary formats, form and content must be separated. Structural / semantic and representational information as well as metadata must be embedded in the text itself in order to be inseparable. We suggest using plain text with rudimentary markup following the conventions of MarkDown and a short block of metadata written in Yaml as the format of choice.

* *Format*: plain text. At their core, all files are simple strings of letters. In the case of plain text files (TXT), this string of letters happens to be human readable. Plain text has been with us since the early days of computing and TXT files could be viewed and edited with 1980s hard- and software. We can therefore assume that this basic format will remain accessible for the years to come.
* *Syntax*: [MarkDown](http://daringfireball.net/projects/markdown/syntax) and its derivatives provide a simple but formal way of providing structural (headings, lists, notes, block quotes etc.) and formating (italics, bold) information that is both human and machine readable. Think of it as the old way of underlining a line of text on a type-writer to indicate a heading.[[2]](#footnote-2)
* *Metadata*: [Yaml (Yaml ain’t markup language)](http://yaml.org/) provides a simple way of including structured metadata (data about data) at the beginning of a text file recording information on authors, title, date etc..
* *Tools*: All major operating systems come with basic text editors sufficient for reading and editing plain text files (“Notepad” on Windows, “TextEdit” on Mac{>> comment on Linux<<}). However, some additional features, such as syntax highlighting and basic customisation of the writing interface (you have to look at it for substantial periods of time), will significantly enhance the writing experience.
  + [Sublime Text](https://www.sublimetext.com/) (Windows, Mac, Linux)
  + [NotePad++](https://notepad-plus-plus.org/) (Windows only)

### 2. ascertaining authorship, version control, and archiving

Writing is a process and subject to change. We need to be able to try out different structures and formulations and, more often than one would like to, we discover that yesterday’s deletions would have been worth keeping. Not to mention an external editor or collaborators that quickly make any approach involving ever-longer file names futile (we all have our folders full of text.docx, text-new-version.docx, text-new-version-2004-01-01.docx, text-new-version-2004-01-01-comments-by-tg-2.docx etc.).

* [*git*](https://git-scm.com/): git is an open *version control system* (vcs) that works with any file type. It traces every change with clear information on author and a timestamp.
* [*GitHub*](https://www.github.com): GitHub is probably the most popular *distributed version control system* (dvcs) and code-sharing platform based on git.[[3]](#footnote-3) While GitHub is a commercial company, it offers free accounts and unlimited public repositories to everyone and free private repositories to academics. Once a change has been committed to a GitHub repository, authors have a public proof of their authorship with a unique identifier that can be referenced. In addition, the repository forms a redundant online back-up of your work.
  + *ssh* (secure shell): in order to communicate with a server without a local client software, some knowledge of ssh is necessary. To avoid this, one can use any of the numerous GitHub clients.[[4]](#footnote-4)
* [*Zenodo*](https://zenodo.org/): Zenodo is an open science platform developed and operated by CERN. Some of its many features are the provision of DOIs ([*digital object identifiers*](https://www.doi.org/)) and long-term storage thus providing stable links and protection against [link-rot](https://en.wikipedia.org/wiki/Link_rot). It also hooks into GitHub.

### 3. collaboration / external review

* git / GitHub: git allows for branching and forking of text. All changes / suggestions can be reviewed and the author can decide whether to accept or reject them.

{>>We need additional collaborative tools<<} - [*prose.io*](http://prose.io/): author text on GitHub online.

## 2. Publishing

### 1. Generate an accessible representation of your text

While it would be absolutely sufficient to publish / distribute the plain-text files by means of a USB key, a graphical user interface (GUI) that translates the structural information into a formatted and aesthetically pleasing layout is often {==advisable==}{>>better wording?<<}—be they a printed paper copy or a website:

* [*Pandoc*](https://pandoc.org/): Pandoc is a tool for converting plain text and markdown into multiple (hence *pan*) target formats (thus *doc*), such as, but not limited to HTML, DOCX, and PDF. Pandoc, which is under active development by John MacFarlane, a professor of philosophy at UC Berkeley, also supports the formatting of references using [BibTeX](http://www.bibtex.org/) (another plain text format for storing structured bibliographic information) and [CSL (citation style language)](http://citationstyles.org/).
* *HTML and CSS*: HTML and CSS are well-established standards to separate form and content maintained by the [World Wide Web Consortium (W3C)](https://www.w3.org/). While the *hyper text markup language* (HTML) carries the content of our text as well as all the structural and formating information and the metadata, *cascading stylesheets* (CSS) provide the actual layout. This combination allows to provide different layouts for different contexts and devices using the same content.
* *Metadata*: HTML5 supports semantic tags and machine-readable metadata following various standards, such as those provided by [Dublin Core (DC)](http://dublincore.org/) or [schema.org](https://schema.org). If one includes Dublin Core metadata in the head of HTML files, aggregators, search engines, and reference managers can find and extract the structured information on author, title, publication date, keywords, etc.

### 2. Think about and provide a licence

{>>mention copyright<<}

A licence is formal agreement that specifies the rights and duties of both the *licensor* (e.g. us as authors) and the *licencee* (e.g. us as readers). Its most important purpose within our discussion is to assure the readers of our texts of their rights to read, copy, and cite them. An open licence might for instance allow reproduction of the text but might prohibit charging for accessing the reproduction.

Formulating one’s own licence text is a challenge and one might not be familiar enough with the necessary “legalese” to write a text readers can rely on. In consequence we suggest looking at established licences and having made a case for open access {>>open science, open knowledge etc.<<}, we suggest to start with [creative commons licences](https://creativecommons.org/).[[5]](#footnote-5)

### 3. Publish / distribute the content

Websites of more than a single page of text tend to be technically complex and require an infrastructure of data storage, internet connections, web addresses, some *content management system* (cms), and databases that rarely come for free and without the need of maintenance. In addition, contemporary dynamic websites are almost impossible to archive or download in their entirety. To reduce complexity and the number of technologies, we suggest using static, self-contained websites without a content management system and no database.

* *jekyll*: [Jekyll](https://jekyllrb.com/) is currently one of the most popular open tools to generate “dynamic” static, self-contained websites from plain text files containing MarkDown and Yaml, using nothing but HTML and CSS and nested files and folders. These can either be hosted online or distributed on a USB key.
* *GitHub pages*: There are, of course, plenty of open, free or commercial providers to host your website. But having already signed up to GitHub and keeping all our texts in a GitHub repository, it is worthwhile to look at [GitHub pages](https://pages.github.com/), which provide free hosting of version-controlled content and supports jekyll out of the box. This means one can directly publish Markdown-formatted plain text files through GitHub pages and jekyll.

# Challenges:

How to deal with sensitive data / material, that should not be publicly accessible, such as ethnographic field notes?

* [OpenPGP](http://openpgp.org/): PGP stands for “pretty good privacy” and is widely used for encrypting emails. OpenPGP is a proposed standard in [RFC 4480](https://tools.ietf.org/html/rfc4880). It works with plain text and thus with all the tools covered in this {==course==}{>>workshop<<}.

# Resources / literature:

* [The Programming Historian](http://programminghistorian.org/): open-access, peer-reviewed suite of tutorials that help humanists learn a wide range of digital tools, techniques, and workflows to facilitate their research.

{>>add relevant literature<<}

## Use of git and GitHub in the humanities

* makerlab at UVic: [git/github in 20 steps](https://github.com/uvicmakerlab/dhsi2015/blob/master/git.md)
* Chad Black’s [getting started with github and prose.io](https://parezcoydigo.wordpress.com/2013/08/26/getting-started-with-github-and-prose-io/) on using a combination of GitHub, jekyll, and prose.io for a collaborative seminar [blog](http://dh.chadblack.net/info/all_posts/).
* [GitHub, Academia, and Collaborative Writing](http://www.hastac.org/blogs/harrisonm/2013/10/12/github-academia-and-collaborative-writing)
* [Push, Pull, Fork: GitHub for Academics](http://www.hybridpedagogy.com/journal/push-pull-fork-github-for-academics/)
* [GitHub for Academics: the open-source way to host, create and curate knowledge](http://blogs.lse.ac.uk/impactofsocialsciences/2013/06/04/github-for-academics/)
* [Living in a Plain Text World (Tools We Use)](http://savageminds.org/2013/02/15/living-in-a-plain-text-world-tools-we-use/)
* [This software for academic paper writing is inspired by Git](http://technical.ly/brooklyn/2014/09/12/authorea-software-for-academic-paper-writing-is-inspired-by-git/)
* [How To: Use Git to Version Your Writing](http://www.lifehack.org/articles/technology/how-to-use-git-to-version-your-writing.html)
* [Using git in my writing workflow](https://www.martineve.com/2013/08/18/using-git-in-my-writing-workflow/)

1. Tenen, Dennis and Grant Wythoff. “Sustainable Authorship in Plain Text Using Pandoc and Markdown.” <http://programminghistorian.org/lessons/sustainable-authorship-in-plain-text-using-pandoc-and-markdown>. [↑](#footnote-ref-1)
2. MarkDown really is only a convention and [John Gruber’s (and Adam Swartz’ [yes, *the* Adam Swartz]) canonical description of the Markdown syntax](http://daringfireball.net/projects/markdown/syntax) is at least partially ambiguous and and lacks some core functionality for academic writing, such as support for tables and footnotes. In consequence, a plethora of formats (MultiMarkdown, GitHub flavored markdown, etc.) and software implementations have proliferated inspired by and based on Markdown that make the actual rendering of Markdown in HTML rather unpredictable beyond the core functionality. In recent years, a group of people involving [John MacFarlane](http://johnmacfarlane.net/), professor of philosophy at UC Berkeley and author of Pandoc, proposed and developed are more rigid standard which they call [CommonMark](http://commonmark.org/). [↑](#footnote-ref-2)
3. Other options based on git are [BitBucket](https://bitbucket.org/) and [GitLab](https://about.gitlab.com/). [↑](#footnote-ref-3)
4. GitHub provides its own clients for [Mac](https://central.github.com/mac/latest) and [Windows](https://github-windows.s3.amazonaws.com/GitHubSetup.exe) [↑](#footnote-ref-4)
5. There is a great list of open source licences, including links to their full texts, at [opensource.org](https://opensource.org/licenses/category) [↑](#footnote-ref-5)