

Publikationswerkzeuge für offene Wissenschaft und freie Software

Christian Pietsch, Vitali Peil, Najko Jahn

im Kolloquium Wissensinfrastruktur

Universität Bielefeld

8. November 2013

Agenda

- Fall A
- WYSIWYG & WYSIWYM
- Auszeichnungssprachen
- Demo
- Fall B
- Fall C

Fall A: Eine Open-Access-Initiative

- [OALI](#): eine Initiative für „Open Access in Linguistics“
- gründet den Verlag [Language Science Press](#)
- Auszug aus <http://hpsg.fu-berlin.de/OALI/setup.html>:

We accept submissions in [LibreOffice](http://de.libreoffice.org/) or MicroSoft Word but strongly prefer LaTeX submissions. [...] [Word to LaTeX conversion](http://hpsg.fu-berlin.de/OALI/word-to-latex.html) is the most expensive part of our way to produce books, so authors can help this project by learning LaTeX. It will improve their workflow and the quality of their work.

Notiz: See [Formats](http://hpsg.fu-berlin.de/OALI/vision.html#formats) on further plans for producing XML and other formats.

Das falsche Versprechen WYSIWYG

WYSIWYG: *What You See Is What You Get*

Beispiele:

- Bürosoftware wie MS Word und OpenOffice/LibreOffice
- DTP-Software wie Adobe InDesign, QuarkXPress und Scribus
- visuelle HTML-Editoren
- offline: Dreamweaver und KompoZer
- online: FCKeditor und TinyMCE

Vorteile:

- Vorschau des Ergebnisses sofort
- flache Lernkurve
- sehr verbreitet

Nachteile:

- mangelnde Portabilität zwischen Computern und Programmen
- mangelnde Haltbarkeit
- vendor lock-in (Abhängigkeit von einem Anbieter)
- Überoptimierung auf eine Darstellung/ein Ausgabeformat führt zu schlechter Wiederverwendbarkeit
- nur wenige User verwenden logisches Markup (Formatvorlagen) zur Strukturierung
- keine vernünftige Versionierung
- keine professionelle Druckqualität (außer DTP-Software)

WYSIWYM: die nachhaltige Alternative

WYSIWYM: *What You See Is What You Mean*

Vorteile:

- Trennung von Präsentationsform und Inhalt durch deklaratives/semantisches Markup
- ASCII und Unicode sind offene, stabile Formate, die sehr viele Programme beherrschen
- ideal für Langzeitarchivierung
- einfacher Editor genügt, z.B. Notepad++, Textmate, nano - oder ein Textfeld auf einer Website
- Teamarbeit + Versionierung (z.B. mit git)

Komplizierte Auszeichnungssprachen

- XML-basierte Auszeichnungssprachen:
- *DocBook*
- *TEI*
- Für Mathematiker, Ingenieure und Naturwissenschaftler: *LaTeX*
- [einfaches Beispiel](#)
- [realistischeres Beispiel](#)

„Hallo Welt“ in DocBook-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<book xml:id="simple_book" xmlns="http://docbook.org/ns/docbook"
      version="5.0">
  <title>Die Welt</title>
  <chapter xml:id="chapter_1">
    <title>Das Hallo</title>
    <para>Hallo Welt!</para>
  </chapter>
</book>
```

„Hallo Welt“ in LaTeX

```
\documentclass{article}
\begin{document}
Hallo Welt!
\end{document}
```

„Hallo Welt“ in Markdown

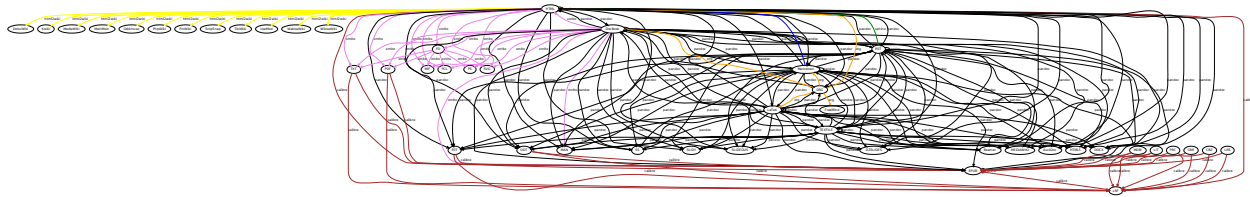
```
Hallo Welt!
```

Unkomplizierte Auszeichnungssprachen

(engl. [lightweight markup languages](#))

- Wikisprachen: *Creole*, *DokuWiki*, *Kwiki*, *MediaWiki*, *MoinMoin/Trac*, *Oddmuse*, *PhpWiki*, *PmWiki*, *SnipSnap*, *TikiWiki*, *UseMod*, *WakkaWiki*, *WikkaWiki*, ...
- *Markdown* bei GitHub/GitLab, StackExchange, Reddit
- *BBCode* in vielen Webforen
- *Textile* in Redmine, Jira
- *Org-mode* im Emacs
- *reStructuredText* eingebettet in Python-Code
- *AsciiDoc* als DocBook-XML-Alternative
- *POD* (Plain Old Documentation) eingebettet in Perl-Code

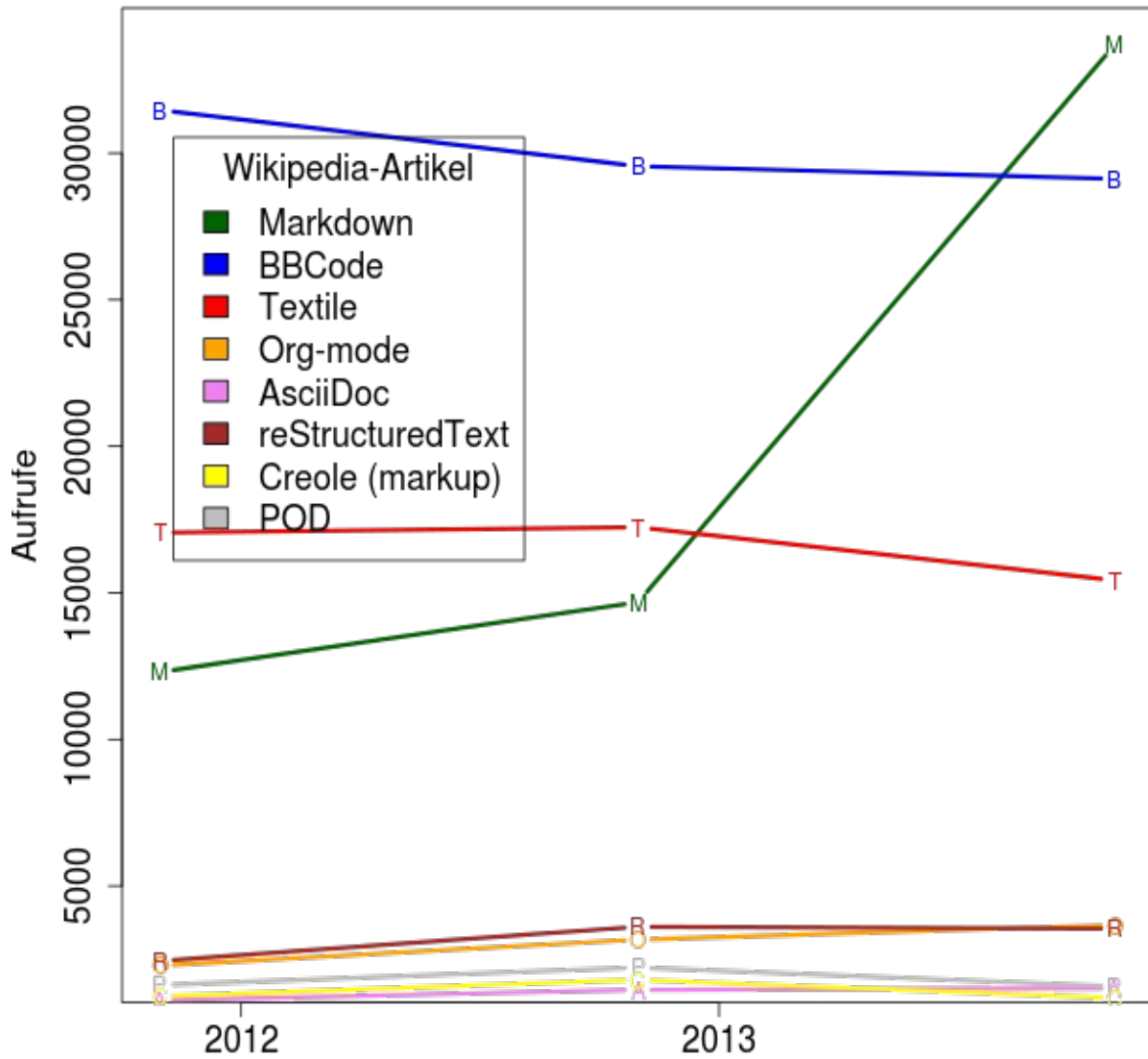
Semantisch annotierte Dokumente lassen sich verlustarm konvertieren



```
digraph format_converters {
  HTML -> {DokuWiki Kwiki MediaWiki MoinMoin Oddmuse PhpWiki PmWiki SnipSnap
          TikiWiki UseMod WakkaWiki WikkaWiki} [label=html2wiki,
color=yellow];
  DocBook -> {EPUB FO HTML MAN MIF PCL PDF PS SVG TXT} [label=xmpto,
color=violet];
  FO -> {MIF PCL PDF PS SVG TXT} [label=xmpto, color=violet];
  HTML -> {FO MIF PCL PDF PS SVG TXT} [label=xmpto, color=violet];
  RST -> HTML [label="sphinx", color=green];
  Markdown -> HTML [label="jekyll", color=blue];
  {MOBI LIT PRC EPUB ODT HTML CBR CBZ RTF TXT PDF LRS} -> {LRF EPUB}
[label="calibre", color=brown];
  ORG -> {FreeMind HTML DocBook LaTeX Markdown} [label="org", color=orange];
  {Markdown RST DocBook TEXTILE HTML LaTeX} -> {HTML HTML5 S5 SLIDY SLIDEOS
DZSLIDES DocBook LaTeX Beamer MAN Markdown RST MEDIAWIKI TEXTILE RTF ORG
AsciiDoc ODT DOCX EPUB} [label="pandoc", color=black];
}
```

Welche wird sich durchsetzen?

Seitenaufrufe im Oktober 2011/2012/2013 in der engl. Wikipedia



Markdown

<http://daringfireball.net/projects/markdown/>

Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).

- Alles sehr simpel. Reicht aber für Romane.
- Verzicht auf Tabellen. Aber HTML erlaubt.

Markdown-Erweiterungen

- MultiMarkdown: Fußnoten, RTF-Ausgabe
- PHP Markdown Extra: Fußnoten, Tabellen, Definitionslisten
- GithubFlavoredMarkdown: Syntaxhervorhebung für Programmcode, ~~Durchstreichung~~, [x] Aufgabenlisten
- Pandoc-Markdown: Syntaxhervorhebung für Programmcode, Tabellen, TeX-Formeln, Literaturangaben mit BibTeX/MODS/RIS und CSL
- Doxygen: Dokumentationsgenerator für C, C++, Java und Python akzeptiert Markdown

Scholarly Markdown

- <https://github.com/scholmd/scholmd/wiki>

Markdown is a fantastic and minimalist tool for authoring scientific documents. This repository is a collection of tools, resources, and tutorials to simplify your workflow. If you spend a little time going through the tutorials you'll be able to stop using Microsoft Word entirely and write clean, lightweight markdown files that can easily be version controlled by git. Collaboration with your coauthors would also become way more powerful and simpler.

Einsatzbeispiele für Markdown

- HTML5-Präsentationen mit Markdown-Input: [Reveal.js](#)
- Universalkonverter für unkompliziertes Publizieren: [Pandoc](#)
- spezielle Editoren: [Qute](#), [ReText](#) oder online [prose.io](#)
- Static Site Generators, z.B. [Jekyll](#)
- Dokumentationsgeneratoren, z.B. [Sphinx](#)

Fall B: Ein Anhang an einem Artikel in Science

<http://nsaunders.wordpress.com/2013/11/06/why-bioinformaticians-hate-the-traditional-journal-article/>

<http://www.sciencemag.org/content/341/6143/281/suppl/DC1>

Beyond the PDF

- statische, seitenorientierte(!) Dokumente sind im Web nicht sinnvoll
- Nachnutzung kaum möglich
- "enhanced publications", "article of the future"
- Publikationen <--> Forschungsdaten

Fall C : Webanwendungen mit R

- [R](#) ist eine statistische Programmiersprache
- hoher Dokumentationsanspruch in den Wissenschaften
- Reportgeneratoren wie [knitr](#), [sweave](#) oder [brew](#)

R an der UB

- Auswertungen von Publikationsdaten
- Vorbereitung von Anreicherungen der Publikationsdaten
- Pakete für wissenschaftliche Dienste, z.B. [rebi](#) für [rOpenSci](#)

Ein Beispiel

Europe PMC Zitationen für den Artikel

Rosso MG, Li Y, Strizhov N, Reiss B, Dekker K, Weisshaar B. An Arabidopsis thaliana T-DNA mutagenized population (GABI-Kat) for flanking sequence tag-based reverse genetics. Plant Molecular Biology. 2003;53(1):247–259. PMID:[14756321](#)

lade Pakete

```
install.packages("devtools")
require(devtools)
install_github("rebi", "ropensci")
require(rebi)
```

Abfrage Zitationsdaten

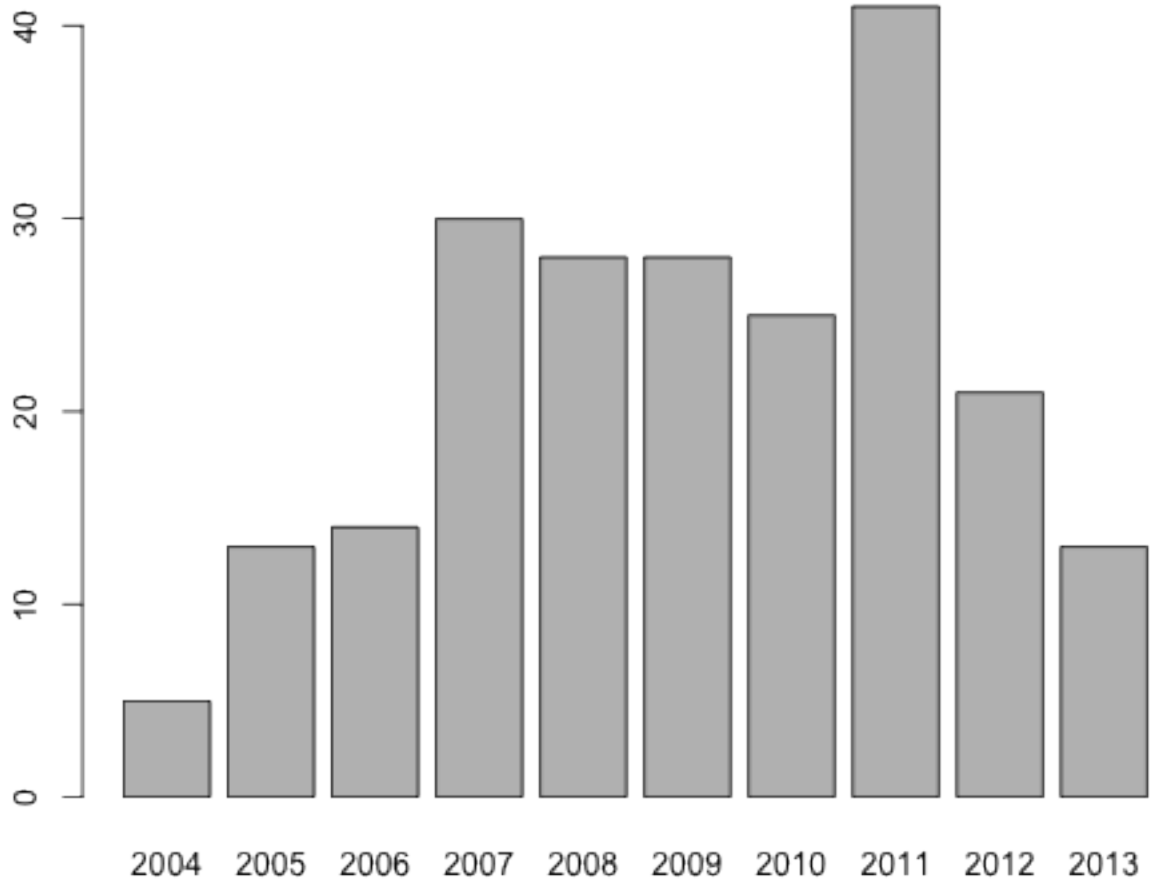
```
my.data <- getcitations(id = "14756321")
```

Die Funktion `getcitations.r` liefert folgende Felder

```
## [1] "id" "source" "citationType"
## [4] "title" "authorString" "journalAbbreviation"
## [7] "pubYear" "volume" "issue"
## [10] "pageInfo" "citedByCount" "text"
## [13] "issn"
```

Balkendiagramm nach Publikationsjahr

```
plot(my.data$pubYear)
```

Häufigste zitierende Journals

```

rev(sort(table(my.data$journalAbbreviation)))[1:5]
##
##           Plant J.           Planta
##           55             13
##   Plant Mol. Biol. Proc. Natl. Acad. Sci. U.S.A.
##           13             12
##   Plant Cell Physiol.
##           11

```

Reportingsprache brew

1· Sämtlicher Text, der nicht innerhalb der folgenden Bezeichner verfasst wird, wird wie in der Eingabe ausgegeben

Reportingsprache brew

2· R-Anweisungen innerhalb von `<%` und `%>` werden ausgeführt

```
<%  
2 + 3  
%>
```

Reportingsprache brew

3· Werte der Funktion innerhalb von `<%=` und `%>` werden ausgegeben

```
## [1] 5
```

Reportingsprache brew

Konvertierung

```
library(brew)
```

```
brew("input.md", "output.md")
```

PUB INSPIRE MONITOR