

Print-optimierte Formatierung wissenschaftlicher Papers mittels HTML

Thema:

Print-optimierte Formatierung wissenschaftlicher Papers mittels HTML

Art:

BA

Betreuer:

Raphael Wimmer

Student:

Maximilian Wagner

Professor:

Christian Wolff

Status:

abgeschlossen

Stichworte:

JavaScript, HTML5, PDF, Paper Templates

angelegt:

2015-06-30

Beginn:

2015-07-20

Antrittsvortrag:

2015-08-17

Ende:

2016-03-31

Hintergrund

Standardisierte Formate wissenschaftlicher Paper können derzeit typischerweise mit Word (o. ä.) bzw. mit TEX erzeugt werden, was die Auswahl an Arbeitswerkzeugen einschränkt.

Zielsetzung der Arbeit

Entwicklung einer Infrastruktur, die mit Hilfe von Webtechniken (HTML, CSS, JS etc.) entsprechende Zielformate automatisch erzeugt und ausgibt (HTML, PDF).

Konkrete Aufgaben

- Recherche und Dokumentation des aktuellen Forschungsstandes zum Thema (Projektarbeit Bertram, Scholarly HTML, „Beyond the pdf“-workshop, bestehende und etablierte Standards weitestgehend berücksichtigen > welche semantischen tags?)
- Entwicklung von Metriken, um die Ähnlichkeit der Druckausgabe mit der Formatvorlage vergleichen zu können (z.B. Differenzbild zwischen zwei Renderings, Liste numerischer

Formatvorgaben überprüfen, automatisierte Überprüfung, Git Repository mit Issues ...)

- Entwicklung eines Templates inkl. CSS & JavaScript für drei relevante Vorlagen (modularer Aufbau, grundlegendes CSS und Script, das für alle Formatierungen gilt, Erweiterung je nach Anforderung der speziellen Formatvorgaben, Verwendung von media queries, Verwendung von semantischen HTML5-Tags, Klassen dann verwenden, wenn es keine ausreichenden semantischen tags gibt, JavaScript so minimal wie möglich fürs Rendering einsetzen)
- Dokumentation von Problemen (Limitierungen der Standards wie HTML5, Unterschiede zwischen den Browsern)
- Testweise jemanden einen existierenden paper-text formatieren lassen und das Ergebnis bewerten (wo traten qualitative Probleme auf? Haben die Leute ein anderes mentales Modell von Bildern? Verwenden sie divs so wie von mir geplant?...)
- Optional: markdown-workflow implementieren und evaluieren
- Optional: Wie kann ein HTML-Dokument noch interaktiver und ansprechender gestaltet werden, wenn man alle Möglichkeiten einer Internetseite ausnutzt
- Optional: Microformats (zusätzliche Informationen z.B. zu EMail-Adressen, Autoren etc., die maschinen-lesbar sind)
- Ausblick: ePaper-viewer, zusätzliche Funktionen

Erwartete Vorkenntnisse

Gute Kenntnisse im MME

Weiterführende Quellen

- Projektarbeit LMU mit vergleichbarem Ansatz (Kontakt: R. Wimmer)
- [Beyond the PDF](#): Movement seit 2011: Suche nach einer Alternative zu PDFs als Standard für wissenschaftliche Paper (Ziel: HTML)
- [Bramstein Typeset](#): Algorithmus zur Ermittlung des optimalen Zeilenumbruchs bei der Textanzeige im Blocksatz und anderen Text-Layouts. Vergleich des TeX-Algorithmus ([Knuth & Plass](#)) mit CSS-Eigenschaften zur Bestimmung von Zeilenumbrüchen
- [Markdown](#): Vereinfachte Auszeichnungssprache, die in ihrer Ausgangsform leicht lesbar ist (Auszeichnungselemente, die auch in *plain text* gebräuchlich sind). Software wandelt .markdown in XHTML um
- [Pandoc](#): Universeller, erweiterbarer Dokumenten-Converter für Markupsprachen (unterstützt [markdown](#) extensions sowie [LaTeX Mathematics](#))
- [PubCSS by Thomas Park](#): Related Work: Sammlung von HTML-Templates und Stylesheets, mit denen wissenschaftliche Texte für Web und Print formatiert werden können
- [Scholarly HTML](#): Richtlinien für wissenschaftliche Texte, die im HTML-Format verfasst werden sollen
- [Semantic Linefeeds](#): Idee, Zeilenumbruch nach jedem Satz sowie nach semantisch zusammenhängenden Satzteilen einzufügen, um Text für Editoren und Versionskontrolle einfacher verarbeitbar zu machen
- [Tangle](#): JavaScript-Bibliothek zur Erstellung von *reactive documents*

From:

<https://wiki.mi.ur.de/> - **MI Wiki**

Permanent link:

https://wiki.mi.ur.de/arbeiten/generieren_standardisierter_paper-formate_mit_hilfe_von_webtechniken?rev=1460213274

Last update: **09.04.2016 14:47**

