notes for unbound book

john c. haltiwanger

my.intro

(in reference to Rakim, who said "it's not where you're from, its where you're at" i'd like to propose a new variation: "it's not who you are, it's where you're at")

auto-didactic programming skillset free software new media theory (easily one of the most important "open source" tools i've ever come across; i'll come to the topic of my thesis in a moment) r11y mindset (has anyone here heard of i18n? it originates from trying to solve a language barrier; thus my contribution 'r11y' - a shorthand for 'revolutionary'; this is not meant that these or any of my thoughts are revolutionary, only that i like to concentrate on problematics with potentially macro-level consequences; any approach that is concerned with its own approach)

+ typography

had known LaTeX as a common libre tool for some time without comprehending exactly what it was.

my friend's college institutionalized LaTeX for typesetting theses and he mentioned the typographical and, further, the referencing management advantages.

later my sister showed me her PhD workflow in LyX.

what you see is...

WYSIWYM is an adjacent paradigm to the venerable and popular What You See Is What You Get.

it is different in the formulation of its focus: implicit semantics over explicit choice.

it abstracts from the underlying layer in an act of re-presentation. in the end it is feeding a typesetting engine that speaks a dialect of a programming language that is over 30 years old.

this hybridization relies on external styles. this is not in itself a problem, but it turns out that extensive stylistic customization can be a tricky deal in LaTeX. i began exploring other options along the same lines and came across ConTeXt. i'll be returning to the importance of this in a second.

grammatic processes

my thesis was a shift in focus from large-scale, peer-to-peer platform design based on social solutions to "piracy".

there were moments in the class that led to revelations on the integrability of programming tools with humanities essay writing. especially in a collaborative sense.

geert lovink proposed a concept for a new kind of journal based on moving beyond old academic publishing paradigms such as traditional peer review and paper-focused issue formats. i emailed him some thoughts about how web interfaces tied to translation layers could potentially enable for experimental peer review systems and cohesive visual design.

the next thing i knew my thesis was about generative typesetting.

the 'what' is generative

so i wrote a thesis whose case study was the typesetting of itself.

the typesetting was conditional: as a primary order, in order to address its own conditions of relevance, the thesis was determined to be set in both HTML and PDF. such a methodology implied automation.

the easiest option is a direct PDF rendering of HTML. which has no hyphenation.

EEErrr!

but even in the first place, who would write their thesis in HTML? (HTML is a machine-oriented semantic markup, making it non-trivial to attain either literacy or instinctual readability; due to time constraints i will not going to go into detail with the history of HTML and machine-oriented semantic markup;)

email as an origin of visually semantic textuality

email was the first time we truly went peer-to-peer. any node that had access to the internet could send and receive emails because the protocol was designed that way. (this, of course, led to heavy spamming.)

email was also a system where one could not rely on physically derived cues for textual emphasis. for instance the repeated etching of ink or double stamping of typewriters could not be represented by the plain-text (read: 'semantic free') environment.

[DO YOU REMEMBER ALL THE SHOUTING?!?!]

this led to systems of expression which relied on textual clues to provide subtextual meaning. the most obvious are all-caps shout-outs, bane of the unintentionally CAPSLOCKED, and emoticons, a set of sigil remixes which inherently and even systematically re-draw received interpretations of text. (the American english "double-quotes," with their physical offshoot air-quotes, are a perfect pre-digital example of this impulse)

less obvious, but still evocative, is the underscore syntax: by wrapping a word or phrase in the '_' character already existing on (American) standard keyboards, a word becomes reconstituted through some integral process between the reader and an evolving digital textuality.

something of what you see is what you get

just as the code-based TeX has failed to catch on against visually-tuned applications such as InDesign, Word, and even LyX, HTML is burdened by it's history as a totalizing semantic ontology. wrapping all semantic meaning in angle brackets tends to flatten our ability to visually distinguish subtext.

this limits the human readability of HTML even as it increases the machine's comprehension and to a large extent this readability problem explains the recent programmer exodus to other markup formats that are easier to comprehend at a glance.

people began to apply the visually semantic developments within email to enable a precursor format from which HTML would be generated. an important example is Markdown by John Gruber and Aaron Swartz, which you haveen seeing throughout this presentation

relevant to publishing

the most successful visually semantic format, in terms of informational impact and widespread use, is MediaWiki.

wikis as a toolset have not proven a very fruitful substrate for producing essays; they have a strange fragility, requiring constant nourishment; but beyond that, they have presentation problems; also i don't find that the visual semanticization flows nearly as nicely

there is no such thing as the perfect pre-format, for the simple reason that typographic workflows bely one-size-fits-all-solutions

however the benefits of programming tools to the writing of essays can be multifold: tracking of changes, contribution mapping, and deadline management (for starters);

humanities, ho!

the good news is that humanities, unlike the math and science that has been the focus of many pre-format specifications, requires relatively few semantic modifiers: sectioning, bold, italics, bullet-points, long quotes, reference management, and figures (which are often images);

the bad news is that right now 'glue' is too often required

though it is possible, it precludes some knowledge of programming the success of wikis, which generally have more complexity than the average humanities essay, prove that it is not impossible for people to wrap their heads around visually semantic textuality

a mutable translation layer

the system i propose is called Subtext. technically it is called a 'mutable translation layer' because it is intentionally vague; it feeds on 'Whatever', meaning that both the semantics it understands and its procedures of dealing with them are transformable

glue ceases to be programming and becomes configuration;

the same semantics can be interpreted multiple ways, for instance to make a PDF for screen or a PDF for print; or an HTML version and a more ebooky ePub; future-proofing against the Next Great Format

Subtext is a practical experiment at striking a new, theoretically interesting, balance of agency between humans, texts, and computers; MS Word privileges the human, HTML privileges the computer: both to the detriment of the text, whic would ideally be situated in a balanced equation between the two;

a re-composable web interface

as a publishing platform, Subtext offers not only typographic possibilities through its unbound format translation, it can also offer new potential for peer review and collaborative writing through its integration of developer tools;

the server can know who you are without telling anyone else; this can lead to live anonymous peer review, conversations occuring next to a text without the power dynamics of names, and even to granular grading of group writing (Subtext tracks who contributed what, while a teacher can grade the contributions before knowing who is responsible)

and these are (probably) just the tip of the ice berg

conclusion

This is far too complex. (but so is any attempt to integrate multiple output formats; Subtext pushes the complexity into the configuration of a workflow, rather than the into the workflow itself;)

We need something simple, materially reflecting the relative simplicity of a humanities workflow.

We also need people to care. (this seems to be a difficult task; luckily there are enough benefits that, once used, will make the old human-centric workflow look more clearly like the remediated typewriter that it is)

Give up the embodied comforts of a practiced (but proprietary) workflow.

Through this we can merge in the toolsets of distributed orogramming, reconfiguring them for our own ends.

After which, we have successfully liberated humanities writing from proprietary control.