

Formatting Process: intersections of generative design and FLoSS typesetting

Author: John Haltiwanger
Student Number: 6100473
Supervisor: Richard Rogers
Program: New Media
Year: 2009-2010
Date: June 6, 2010

Thesis Chapter List

1. Introduction

- Re-mediation: the drive towards “greater authenticity and immediacy of presentation” (Bolter 2001: 70)
- Historical developments of the various formats (Typewriter -> Word, SGML -> HTML, Typesetting math -> TeX)
- Contrast the approaches: WYSIWYG, semantic markup, formal markup
- the relationship between computers and language (Cramer 2001)
 - ★ computers run on alphabets; “Literature is therefore a privileged symbolic form in digital information systems.” (Cramer 2001: 2)
 - ★ the rule of searching for a specific text phrase is complicated by formats: intermediate layers are required for text searching when ODT/PDF are involved
- the re-remediation of Project Gutenberg: the availability of ASCII (itself a remediation) allows for translation across interfaces; the appearance of the tablets/e-readers as interfaces to the ASCII re-mediate them back into a book

2. A history of cross-media publishing and generative design

- Simondon’s *transduction*: “By transduction we mean an operation—physical, biological, mental, social—by which an activity propagates itself from one element to the next, within a given domain, and founds this propagation on a structuration of the domain that is realized from place to place: each area of the constituted structures serves as the principle and the model of the next area, as a primer for its constitution, to the extent that the modification expands progressively at the same time as the structuring operation” (Simondon 2009: 11).
 - ★ the “pre-format” as preindividual (but still individual); ontologies on top of ontologies
- History of the term ‘media,’ it’s origins in advertising as they developed a language to discuss ‘mediating’ messages, that is, tailoring the message to suit a particular ‘medium’

- Advantages and disadvantages of generative design.
 - ★ Issues of scale can considerably affect the suitability of a generative workflow over a traditional one.
- The Holy Grail: “one system that serves as the universal document source”
 - ★ Resembles other holy grails, “artificial intelligence” and “real-time collaborative editing of the same document”
- Wrappers: software for remediating formats
 - ★ the ultimate intersection point
- Showstoppers: errors and/or limitations in functionality that force abandoning one approach for another
 - ★ limitations of bibliographic intersections
- The role of FLOSS in generative design
- bootstrapping - Engelbart’s idea of using computer tools to make better computer tools

3. Approaches to digital typesetting

- Print is Static, Code is Process: the physics of text
 - ★ Hayles requests taking into account physical specificity; the last chapter will include memory heap visualizations in order to interrogate the utility of such an approach
 - ▷ what about the physicality of the writer in crafting the text? this leads directly into questions of workflow
- WYSIWYG: the computer as a typewriter
 - ★ Concrete poetry and “free form typography” as unique to this approach.
 - ★ “A typewriter (or a computer-drive printer of the same quality) that justifies its lines in imitation of typesetting is a presumptuous, uneducated machine, mimicking the outward form instead of the inner truth of typography.” (Bringhurst 2008: 28)

- Semantic markup: interpreted plaintext (top-down)
 - ★ Tuned for remediation, strict separation of display from content.
- Formal markup: the document typesets itself (bottom-up)
 - ★ The long and varied history of TeX.
 - ★ Why ConTeXt?

4. A Generative Methodology

- Is this process useful? Does such a practical approach have anything to offer theory? And does theory offer anything useful for the practice?
- Visualizations: memory heap analysis (again, is it useful? does it map a ‘materiality’?)
 - ★ also, version control visualizations of the git repository that hosts the thesis files
- Case studies
 - ★ HTML
 - ★ OpenOffice.org
 - ★ ConTeXt

5. Conclusion

Material

Elements of essays

- Formats (MLA, APA, Chicago, etc.) [*European examples? Screen-specific examples?*]
- Annotations (Static (page) vs Dynamic (screen))

Elements of texts

Surface Level

- Hypertextuality (formal or semantic) [*Can a claim be made that Hayles' printed hypertext is semantic? could there be a parallel connection between the surface and subtext layers?*]
- Typographic qualities ("simple issues" of kerning, line breaking; "institutional" formats)

Subtext Level

- 'Setting standards' (WYSIWYGs and markups as "hairy objects")
 - markups are sub-divided into formal (TeX) and semantic (HTML)
 - the file won't automatically look the way you want it (Office on different platforms, OO.o the same; the 'tweak' factor of TeX; browser specific line and character spacing)
- Display-ability (file formats; DRM; the display itself)

Substrate Level

- hardware (rarely discussed, barely understood)
 - but perhaps not necessarily all that *relevant* to the material analysis of text?

- the unspoken labor of *making* that hardware
 - Pasquinelli's parasitic resource extraction

Threads (Introduction)

There are several threads that I perceive as necessary for the construction of a proper report on the subject. The first is the theoretical geneology of the medium. What claims have been made about the *composition* of mediums? The second thread involves interrogating these claims in light of the nature of generative processes. What place do discussions of mediums and medium-specificity have in the realms of morphic generativity? The third thread revolves around documenting the historical dimensions of electronic typesetting. What currents flow through the development of the typesetting tools found today in the world of FLoSS?

The importance of this question lies in the mercurial context of open source, and even computing in general, in which time is spent on coding software rather than writing history. Constructing a historical record is thus an attempt to contribute this relatively invisible undercurrent to the scholarly record of new media in hope of providing a template for further introspection in other fields of software evolution.

The fourth thread involves direct integration of generative processes in a typesetting workflow composed of FLoSS softwares. Through this activity I hope to materially instantiate a zone for discursive interplay between new media theory and real-world typographic engineering. Does theory offer methods or tools useful for interpreting the dynamics of typographic workflows, especially as they involve generative processes? One obvious tool is a vocabulary through which theory can attempt to describe and critique the software's capacities and interactions. This thread of the project aims to deliver exactly that.

Code

The relationship of code to language is that of a subset constrained by the specificities of syntax (Cramer 2001). The digital computer is ruled by syntax, which could be considered the defining means of mediation between digital computers and human processes. These processes include both the actions of the users and the objects created, stored, distributed, and displayed on digital computers.

Florian Cramer identifies language as a “privileged symbolic form” within the context of digital information processing (2001: 2). This privilege results from the fact that the digital computer itself operates on an alphabet—the 0s and 1s feeding the processor in intervals measured in nanoseconds.

Galloway points to the entirely artificial differentiations between ASCII text on a web page and the ASCII text itself from which the web page is rendered (Galloway 2010). This is the point where distinctions between code, formats, and interfaces begin blur. Code is the mechanism which enables formats. Formats in turn mediate their contents, rendering (and often rendered by) different interfaces. The code is an interface to viewing formatted content, and the format is the interface through which the content is presented. The intersection of these elements displays the inherent slipperiness that undergirds our understanding of ‘media’ at a time when more and more media are being subsumed by and remediated onto digital devices.

Software Design

There are definite limits when it comes to constructing software. Beyond the obvious limitations imposed by computing capacities and architectures lie sticky issues of development. In his seminal *The Mythical Man-Month*, Frederick Brooks describes boundaries for software construction that seem to betray logic. Adding money and manpower, for instance, do not increase productivity. In fact, there is a real chance that these types of influx will accomplish the opposite effect (Brooks 1975).

The errors of managing software construction that Brooks describes appear to be endemic. In his book *Dreaming in Code*, Scott Rosenberg documents the impact of these boundaries of human process on the development of Chandler, an open-source personal information manager (PIM) backed by millionaire code celebrity Mitch Kapor (Rosenberg 2008). When announced in the fall of 2002, Kapor stated that “optimistically” the project would reach a 1.0 release by the end of 2003, while a pessimistic projection would place

such a release in 2004 (83). At the time of *Dreaming in Code*'s release in 2007, Chandler had yet to achieve that milestone. Only in 2009 did the software finally achieve this milestone.

Architectures of FLoSS Typesetting

WYSIWYG

Professional typesetting today is largely accomplished within the proprietary program InDesign from design software superpower Adobe, Inc. InDesign provides a WYSIWYG (What You See is What You Get)^[1] interface that allows direct visual manipulation of documents.

Semantic Systems

Formal Systems

[1] : The acronym is generally pronounced “wizzy-wig.”

Generative Design

- “interfaces” libraries, formats, and languages to create a “medium” through which to generate an “output”
- can be self-perpetuating, ala demoscene
- requires environments of relative specificity
 - can resemble alchemy in necessitating very strict, even ‘esoteric’ steps/connections
- Simondon’s transduction: “By transduction we mean an operation—physical, biological, mental, social—by which an activity propagates itself from one element to the next, within a given domain, and founds this propagation on a structuration of the domain that is realized from place to place: each area of the constituted structures serves as the principle and the model of the next area, as a primer for its constitution, to the extent that the modification expands progressively at the same time as the structuring operation” (Simondon 2009: 11)

Generative practices increasingly take a larger prominence in design workflows. “The process is the product,” declares the *Conditional Design Manifesto*. Employing the “methods of philosophers, engineers, inventors and mystics,” the four authors of the manifesto seek to abandon the idea of a product in favor of “things that adapt to their environment, emphasize change and show difference” (Maurer, et al.). The text is split into a prose introduction and three sections of manifesto-like declarative sentences:

The powerful Java-platform tool named Processing has become wildly popular amongst visualization designers. The demo scene has long employed generative techniques to instantiate complexity on-screen using as few as 4 kilobytes of compiled binary code.

The case study at the core of this paper will explore these concerns as the materiality of the thesis itself remains ethereal. It is more than its input, it is more than its output: it is literally and figuratively very much a *process*.

The substrate of the text you are reading right now is a simple formal markup language called Markdown. Originally written in Perl, it has spread to any language that ends up touching the web. This is due to its popularity as an approachable ‘pre-formatting’ markup language for operations such as blog posts and comments, where HTML is the desired output but a simpler *and more readable* input is often preferable.

Formats

Formats are function calls. They require standards and standards-based libraries. There is no point to a format if there is nothing there to decode it. Nihilism, however, provides no proper formats—nothing is worth the effort of decoding. So issues of ‘nothing’ and ‘something’ are complicated by the mercurial nature of software. There are an infinity of inbetweens on the way to an invisible interface.

Formats pre-date digital computing. The realm of academic publishing is awash in formats: Chicago, APA, MLA, etc. These formats are achieved via compositional guidelines. These guidelines are constructed with consideration to issues ranging from ink use to politics. [need more history of these formats]

Formats as Databases

[Pending]

Bibliography

If the format is a database, what of the fact that it can’t accurately describe itself without human intervention? Binary files are content-opaque. Even text files require some form of semantic standard if their contents are to be understood without dedicating time to mining the text for contextual clues. This begs the question of bibliography.

Bibliography is the most visible difference between the formats of academia. It is also a point of regression for the tri-mediation project that brings this text to the screen or page you are reading it on. This is despite the fact that both OpenOffice.org and ConTeXt have, through third party modules, facilities to greatly reduce the tedium and complexity of citation.¹ The wrapper software pandoc constrains me to self-managed citations because it does not integrate bibliography into its own workflow: it simply passes the TeX command `\cite{}` to the next stage in the toolchain, providing that next stage is intended for TeX. If any other file format is used, the `\cite{}` command is ignored.

Degrees of Volatility

- Text encodings: the complicated nature of bit-constraints. The introduction of Unicode as a solution and the time it has taken to achieve UTF ubiquity.

¹ In the case of OO.o, there is Zotero (<http://zotero.org>) and in the case of ConTeXt there is the bib module () as well as the possibility of integrating any other TeX-compatible bibliographic module through some degree of individual effort.

- File formats can be vectors of infection. Look at recent security notifications coming from Adobe. Longstanding MS Office vulnerabilities.
 - Free Software offers accountability; the source should not be vulnerable for long after an exploit is unveiled.

Interface

Galloway offers a tentative definition of interface: “the interface is defined as the artificial differentiation of two media” (Galloway 2010). The interface can be seen in light of Dagognet’s concept of a fertile nexus,

Popularity as the Ultimate Tractor App

In some ways popularity is the granular mechanism of achieving an interface that is simply used, not seen. That is to say, the relative ubiquity of an interface in part defines whether that interface appears ‘natural’ or not.

The entire levelling of standards within typographic design evolved in the context of readability. What is attractive to the eye? How do you ‘encode’ messages through the formatting of text on a printed page in a way that is striving towards, and at least to some degree achieving, invisibility?

It is no secret that for a long time text sought invisibility in its conveyance. That is a romantic ideal which has stuck with us. The author is privileged, perhaps in no small part to the physicality of a text’s construction. What can lead to a finished state of readability? The audience can never know beyond its aesthetics (including its textual aesthetics, that is to say its “wording”).

Concrete Poetry

Concrete poetry developed through the relations of certain human processes with the interface of the typewriter.

Electronic Typesetting

Pre-GUI options

Word Processors

NBI - Nothing But Initials

One person in charge of knowing a word processing system would type manuscripts for teams or even entire departments.

Edix-Wordix

ConTeXt

ConTeXt is the smaller of the two notable macro packages in use as of 2010. Developed by the Dutch publishing house Pragma ADE, ConTeXt's scope is at once narrower and more flexible than the vastly more popular LaTeX.

Document classes vs. Environments

ConTeXt's name is, like many software projects, a pun. This pun revolves around the concept of *cons* that features prominently in

Bibliography

Bolter, Jay David. (2001). *Writing Space: Computers, hypertext, and the remediation of print*. New York: Routledge. Print.

Bolter, Jay David and Richard A. Grusin. (1996). "Remediation". *Configurations* 4:3. PDF.

Bringhurst, Robert. (2008). *The Elements of Typographic Style, version 3.2*. Vancouver: Hartley & Marks. Print.

Brooks, Frederick. (1975). *The Mythical Man-Month: Essays on software engineering*. Massachusetts: Addison-Wesley. PDF.

Cramer, Florian. (2001). "Digital Code and Literary Text". *netzlituratur*. Web. <http://www.netzliteratur.net/cramer/digital_code_and_literary_text.html> (last accessed 5 June 2010).

Cramer, Florian and Matthew Fuller. (2008). "Interface." In *Software Studies: a lexicon*, edited by Matthew Fuller. MIT Press: Cambridge.

Fuller, Matthew. "It looks like you're trying to write a letter: Microsoft Word". 2000. Web. <<http://www.nettime.org/Lists-Archives/nettime-l-0009/msg00040.htm>> (last accessed 5 June 2010).

Galloway, Alexander. (2010). "Interface". Presented at *A wedge between public and private conference* on 22 April 2010, Amsterdam.

Gitelman, Lisa. (2008). *Always Already New: Media, history, and the data of culture*. Cambridge: MIT Press. Print.

Hagen, Hans. (2009). *The history of luaTeX*. Netherlands: Pragma ADE. Web. <<http://www.pragma-ade.com/general/manuals/mk.pdf>> (last accessed 5 June 2010).

Hayles, N. Katherine. (2004) "Print is Flat, Code is Deep: The Importance of Media Specific Analysis". *Poetics Today* 25:1. PDF.

Hayles, N. Katherine. (2008). *Electronic Literature: new horizons for the literary*. Notre Dame: University of Notre Dame. Print.

Kinross, Robin. (2004). *Modern typography: an essay in critical history*. London: Hyphen Press. Print.

Knuth, Donald. (1984). *The TeXBook*. Massachusetts: Addison-Wesley. Print.

Knuth, Donald. (1999). *Digital Typography*. Stanford: CSLI. Print.

Manovich, Lev. (2001). *The Language of New Media*. MIT Press: Cambridge. Print.

- Rosenberg, Scott. (2008). *Dreaming in Code*. New York: Three Rivers Press. Print.
- Simondon, Gilbert. (2009). "The Position of the Problem of Ontogenesis". Gregor Flanders, trans. *Parrhesia* 7. Web. <http://www.parrhesiajournal.org/parrhesia07/parrhesia07_simondon1.pdf> (last accessed 5 June 2010).
- Snelting, Femke. (2009). "The Making of". *Tracks in electr(on)ic fields*. Brussels: Constant. Web. <http://ospublish.constantvzw.org/wp-content/uploads/makingof.pdf> (last accessed 5 June 2010).

Generative Re-mediation

An investigation of cross-media publishing in free and open source software

Student: John Haltiwanger
Paper: Final Thesis Proposal
Supervisor: Richard Rogers
Date: 6 March 2010

Introduction

The very term ‘media’ comes from the discipline of advertising, which in the 1940s began to develop a vocabulary to discuss the emerging issue of delivering messages across various distribution channels. While this initial impulse did not necessarily involve an algorithmic capacity to target multiple mediums using a single input, the advent of networked computers has seen just such a desire arise. ‘Traditional’ design tools such as the Adobe Creative Suite are being replaced in top design firms with workflows that dynamically generate documents. Rather than constructing everything by hand in what the What You See Is What You Get (WYSIWYG) interfaces of those programs, designers are now using software *wrappers* and scripts that facilitate translations of one format into another. Tied together in a deliberate workflow, the tools receive relatively simple source documents as input and produce multiple outputs given the parameters the designer has implemented into their workflow. This process is called *cross-media publishing*.

Free, Libre, and Open Source Software (FLOSS) often plays an important role in such workflows. FLOSS programming languages, wrappers, frameworks, toolkits, and applications are increasingly incorporated into a design workflow. The success of the generative art toolkit Processing is exemplary in this regard¹. The WYSIWYG application Scribus, which aims to provide an interface with capabilities analagous to Adobe InDesign, provides Python-based scripting functionality to allow a hybridized approach. Indeed, Adobe has developed an XML-based input format for InDesign that allows generative production of InDesign documents. Cross-media publishing and generative techniques are not only increasingly successful in the design world, they are also closely related. This thesis aims to investigate this relationship through a specific mode of cross-media publishing. Using a specific wrapper called Pandoc², the thesis will be

¹ Processing. <http://processing.org/>.

² Pandoc. <http://johnmacfarlane.net/pandoc/>

simultaneously typeset in three output formats: HTML, OpenDocument (ODT), and ConTeXt. Each of these formats represent a distinct approach to the issue of typesetting—these approaches are worthwhile to investigate, as they represent the three approaches to typesetting currently found on the computer: semantic markup, formal markup, and WYSIWYG. The Holy Grail of cross-media publishing, according to Florian Cramer, is “one system that serves as the universal document source.”³

Remediation

The theoretical framework to be deployed in this investigation of generative, cross-media typesetting is Jay David Bolter and Richard Grusin’s concept of remediation (Bolter & Grusin 1999). As defined by the authors, remediation is “the representation of one medium an another” (45). Bolter and Grusin identify two distinct approaches for messages in a given medium. In the first, immediacy, the message is tailored in such a way as to, ideally, make the audience forget the presence of the medium. In the second, hypermediacy, the opposite is sought: unique aspects of the medium are highlighted by the message. Remediation, then, involves an interplay of these two approaches. That the WYSIWYG style developed for Word (and wholeheartedly adopted by OpenOffice.org) takes quite seriously the goal of mimicking a typewriter is then an example of immediacy. Even the approach of representing a page of paper on a screen is an example of remediation. Since HTML and ConTeXt can both be used to create not only remediated paper documents but also ‘screen-native’ pages as well, the theoretical framework of remediation, immediacy, and hypermediacy provides a compelling means for developing this thesis.

Actor-Network Theory

At the risk of mis-applying Bruno Latour’s Actor-Network Theory (ANT), this thesis will incorporate the language and conceptual tools presented in Latour’s *Reassembling the Social* (Latour 2005) in order to properly position the various applications, frameworks, and output formats that will collectively generate and encompass the final output of the thesis. Not only is pandoc a mediator in this thesis, the output formats themselves are as well, as their functionalities, peculiarities, and limitations will have an effect on the content of the thesis as well as its final presentation. The programs and scripts that come into play can also be seen as having mediating roles. I am also an obvious actor in this network, but in more ways than simply writing the thesis—for example, questions I ask to the ConTeXt mailing list are translations⁴ between actors. The

³ From a personal interview.

simple production of this thesis, bridging as it does both free software and generative design, could be thought of as a point of contention through which once unconnected actors are brought into contact with one another. Acknowledging this fact up front and directly through an ANT framing of the project can only strengthen its potential to one day become a mediating text itself. Certainly it seems appropriate to begin to situate softwares as important actors in their own right, especially in this context where the very materiality of the thesis (let alone its content) is dependent on the translations produced by these actors.

Questions

- What is the current state of generative design? What drives its adoption by designers? What role does FLOSS play?
- What are the individual merits of each of the three open source output formats? How do they contrast in relation with each other? What are the output formats unique capacities? The figurations of their capacities emerging from this document are to be discovered in the process of attempting to perform relative tasks between them.
- What capacity do the output formats have for addressing issues of remediation? That is, does a given output format allow for targetting either immediacy, hypermediacy, or a hybridization? Or is only one or the other possible in that format?
- What limitations are present that prevent accomplishing certain typesetting goals in a given format? What limitations in the software tools used will affect and limit the capabilities of this thesis?
- Is it possible, through this approach, to generate aesthetically pleasing typesetting in any of the three formats? What advantages does a ‘single source input format’ have?

Approach

What follows is an example of how the investigation the inclusion of hypertext in a Con-TeXt document can provide a springboard for investigating material differences in the typesetting options.

⁴ In Latour’s words, though I may adopt the less confusing ‘modulation’ as well as ‘assemblage’ over ‘network.’

For instance [this url](#) was input according to the syntax [found here](#). In the Markdown format this code appears as

```
For instance [this url](http://wikileaks.org/) was input according to the
syntax
[found here](http://daringfireball.net/projects/markdown/syntax).
```

When the text `pandoc -t context -o thesis.tex thesis.markdown` is input into a shell prompt, the resulting output appears as follows

```
For instance “useURL[1][http://wikileaks.org/][][this url]“from[1] was input
according to the syntax
“useURL[2][http://daringfireball.net/projects/markdown/syntax][][found
here]“from[2].
```

However, that in itself is not enough for context thesis to generate a valid PDF with hyperlinks. For three additional lines are required.

```
“setupinteraction[state=start,color=darkblue]
“starttext
For instance “useURL[1][http://wikileaks.org/][][this url]“from[1]
was input according to the syntax
“useURL[2][http://daringfireball.net/projects/markdown/syntax][][found
here]“from[2].
“stoptext
```

The differences in the syntax are clearly explained by their status as different markup languages. The structural differences, however, such as the indication of a ‘start’ and a ‘stop’ to ‘text’ as well as a slightly cryptic ‘setup’ for ‘interaction’ reflect that TeX is a *batch text processor* and the way that ConTeXt structures its TeX macros, respectively. In other words, that these additional commands are required reflects architectural differences that in turn reflect the goals around which the text processing software was designed. This is obvious when one considers that the output of `pandoc -t html -o thesis.html thesis.markdown` not only shows HTML’s own structural choices (including its much less awkward hyperlink syntax), it will likely also render in a browser without the addition of HTML’s version of ‘start’ and ‘stop’ (`<HTML>` and `</HTML>`) due to architectural decisions relating to backwards compatibility and leniency belonging to the browser’s design.

Additional work would be required to achieve a decent looking typesetting of the source code blocks above if the Pandoc utility did not already do some primitive line-breaking of its own. ConTeXt has trouble with handling long unbroken strings such as URLs and even text commands, at least in the relatively primitive version of a source code block that is being used here (“starttyping and “stoptyping) as seen in the second code block. Meanwhile the same code blocks look acceptable in ODT and HTML by default, while the ConTeXt output fails as it leaves off portions of text. What procedures must be followed to achieve an acceptable output in ConTeXt? Are they worth the effort? While the ODT output is passable, to what extent can it be improved? HTML is only some lines of CSS away from outputting the complete text of the code blocks with the nicety of an boxed outline to offset it from the general flow.

Expected Findings

1. There will be significant limitations in each format. ODT will be frustratingly manual in its requirements for effecting the desired typesetting. HTML will likely fall short in line-breaking and print quality. ConTeXt will likely prove highly temperamental and difficult to utilize to its full extent.
2. ConTeXt will be the most suitable for straddling ‘immediacy’ and ‘hypermediacy,’ as it allows for targetting both the printed page and the computer screen. It’s superior typesetting algorithms will give it distinct aesthetic advantages, but in the end these will be weighed against the effort expended to achieve the desired result.
3. Generative design workflows will generally involve a majority of FLOSS software.
4. Typographic programming is not a well-documented field. A theoretical-technical of three FLOSS formats in typographic terms will provide a distinct contribution to both traditions.
5. A ‘single source input format’ that is a) plaintext and b) simple and readable is an integral requirement for the collaborative writing of texts.

Bibliography

- Bolter, Jay David and Richard Grusin. *Remediation*. Cambridge: MIT Press. 1999. Print.
- Latour, Bruno. *Reassembling the social*. Oxford: Oxford University Press, 2005. Print.

Formatting Processes: A reflexive investigation of generative typesetting using FLOSS

John Haltiwanger

15 April 2010

Core concerns

This paper claims that the type of material analysis in which Hayles engages is an inadequate formulation of a valid concern. *Scriptons* and *textons* are two terms proposed by Esper Aarseth and utilized by Hayles to refer to the surface of text and the underlying codes that produces that text, respectively. Confusingly, if some degree of code is available for investigation, that code is considered part of the *scripton* (Hayles 2004: 78).¹ This complicates the capacity for discussion as the code that is most relevant for the presentation of text on a screen in any given *specific* instance is necessarily the source code. If the source code is considered part of the ‘surface text’, there is little left to truly investigate on a per-document basis. As important an observation it is that all electronic text begins as voltage signals, it is not much more informative than stating that all printed text begins as ink.

One goal of this paper is to push the boundaries of material analysis to include a detailed examination of surface text and the code used to generate it. In order to do so this paper will focus on the relationships of inputs and outputs in a generative design workflow, specifically the typesetting of the thesis itself. The variance of the chosen output formats in both typographic quality as well as compositional elements allows the interrogation of electronic text on a deeper level. This is not meant to be a simple features analysis. Rather, the goal is to develop a theoretical framework for discussing the separate layers of *text*, *code*, and *type* (as in typography). Without considering issues of presentation (type) or composition (code), a material analysis remains incomplete. In other words, there is a lot of room in Hayles’ “frothy digital middle” for a deeper level of discussion (Hayles 2004:

¹ This isn’t to mention the confusion, from a programming perspective, of using the root words ‘script’ to refer to surface level and ‘text’ to refer to programmatic layers beneath that surface. This is counterintuitive to a tradition in which ‘scripts’ are written to deal with ‘text’ algorithmically.

75). This froth will be examined in light of the remediation theory provided by Jay David Bolter and David Grusin.

Research Questions

- What claims are made about the qualities of a ‘format’ in relation to a ‘medium’? Within computers, to what extent can a specific format yield the same effects/components of a medium?
- What does a material analysis of a *generative process* look like? If a text is merely an input in a process that yields specifically different outputs, what kind of materiality does it have?
- If typography has a material effect, on which layer does it operate? Should typography be incorporated into the material analyses of documents in the tradition of Hayles?
- What claims are made about computer typography? and what claims, if any, are specific open source software?

Generative design

Generative practices increasingly take a larger prominence in design workflows. “The process is the product,” declares the *Conditional Design Manifesto*. Employing the “methods of philosophers, engineers, inventors and mystics,” the four authors of the manifesto seek to abandon the idea of a product in favor of “things that adapt to their environment, emphasize change and show difference” (Maurer, et al.). The text is split into a prose introduction and three sections of manifesto-style declarative sentences: *process*, *logic*, and *input*. These are the governing principles of their proposed angle towards design.

By focusing on process, the authors of the manifesto employ a conscious decision away from productized design. Process produces “formations rather than forms” and facilitates a search for “unexpected but correlative, emergent patterns” (Maurer, et al.). Logic is invoked as their method for “accentuating the ungraspable” and provides the means for designing the “conditions through which the process can take place”. Input is their “material,” and should come from the “external and complex environment”. It “engages logic and influences the process”

The powerful Java-platform tool named Processing has become wildly popular amongst visualization designers. The demo scene has long employed generative techniques to instantiate complexity on-screen using as few as 4 kilobytes of compiled binary code. Video

game design guru Will Wright's 2008 game *Spore* relies heavily on procedural generation. Many open-source (and, presumably, otherwise) projects generate their documentation using a combination of algorithm and in-code markup.² Beyond this, Dutch designer Petr van Blokland has ceased using Adobe products in the design pipeline at his firm. By *programming* instead of *hand-crafting* Blokland is able to achieve a balance of control and flexibility that can support the kind of enormous workflow demanded by clients such as Rabobank.

The case study at the core of this paper will explore these concerns as the materiality of the thesis itself remains ethereal. It is more than its input, it is more than its output: it is literally and figuratively very much a *process*.

The substrate of the text you are reading right now is a simple formal markup language called Markdown. Originally written in Perl, it has spread to any language that ends up touching the web. This is due to its popularity as an approachable 'pre-formatting' markup language for operations, such as blog posts and comments, where HTML is the desired output but a simpler *and more readable* input is often preferable. Using a 'wrapper' tool called **Pandoc**, the Markdown input is converted to HTML, ConTeXt, and OpenOffice.org Writer (ODT) formats. Through customizable templates, **Pandoc** provides a great deal of flexibility to the end-user.

Caveats

Florian Cramer offers a techno-musical analogy for generative design in the context of typesetting: TeX is to WYSIWYG typesetting as a player piano is to a piano under the fingertips of a real pianist (Cramer 2010). Cramer also Femke Snelting, of Open Source Publishing in Brussels, has mentioned "showstopper" issues facing the enactment of specific design goals in OSP's first project using ConTeXt (Snelting 2010).

Theoretical Perspectives

Without having a great deal of literature on "formats" to work with, I would like to assert that formats are sub-species of mediums. The sitcom is a television format, the action blockbuster a cinematic format, and the concrete poem a writing format. It is perhaps within the medium of the microcomputer, however, that the largest number of formats can

² Indeed one of the available input formats (still) in consideration for this thesis is reStructuredText, the markup syntax which lies at the core of the Python programming language's online documentation.

be found. As a fully programmable medium, the microcomputer facilitates the development of an enormous number of formats. These formats can carry political and social significance, as in the case of HTML 5 and the debate over the use of the patent-encumbered h.264 codec versus the open-source, royalty free Ogg Theora format. In the case of computers, then, it is perhaps possible to assert that formats have a more distinct, medium-like dimension than formats in other media have.

In his piece “Media Cold and Hot”, Marshall McLuhan argues that a medium’s capacity to capture a sense defines, in some ways, its social effect. The state of the society to begin with is critical and will determine the nature of a new medium’s impact. First it is important to establish what his scale is based on and how he establishes it. “A hot medium is one that extends one single sense in ‘high definition’. High definition is the state of being well filled with data” (McLuhan 1964: 24). Hot media require little participation because they are so engrossing—their “high resolution” (to switch his phrasing a bit) doesn’t provide or require extensive participation. Cool media, on the other hand, require a great deal of audience participation.

In McLuhan’s view, nationalism and religious wars in sixteenth century Europe were the result of a “hotting-up” of the medium of writing as it transitioned to print (25). Furthermore, “a tribal and feudal hierarchy of traditional kind collapses quickly when it meets any hot medium of the mechanical, uniform, and repetitive kind” (26). In this way media can be seen as highly influential, having the power to completely alter societies, as the introduction of the letter press and the subsequent collapse of Catholic hegemony in various parts of Europe.

Materialist Format Analysis

One of the primary objectives of this paper is to force integration of typographical design principles into a material analysis. Robin Kinross’ argues that the first typographically modern typesetting began much earlier than what is now considered to be the modern era (Kinross).