

The Prime Designer

Jason Lewis Concordia University

a response to Johanna Drucker's "Visual / Digital / Poetical"
Textologies Interdisciplinary Workshop on Multimedia Technologies, Ethics and Culture
McMasters University, Hamilton ON (October 2004)

I have been following Prof. Drucker's work over the last ten years, ever since I began my graduate work on computationally-driven interactive and dynamic texts. Her core theoretical project of illustrating and illuminating the ways in which the material manifestation of a text participates in the meaning-making of that text has served as a touchstone for my own thinking about how—if—the same sort of material analysis applies to the ephemera of pixels painted on the screen and manipulated through code. Finding her project extended so fully into the digital context has been heartening and instructive. I appreciate the opportunity to comment on it.

Prof. Drucker's paper, "Visual / Digital / Poetical", presents a dense mixture of theoretical and practical issues related to the classification of artist-books and works of visual poetry. On the practical side, she is concerned with how we might go about creating a computationally-tractable language capable of capturing the broadest range of phenomenological activation adhering to such works. In order to create such a language, one needs a model through which its operations can be defined. And in order to create a suitable model, one must engage with the theoretical question of, in her words, "how materiality works".¹ This question generalizes across all textual artifacts, as Prof. Drucker notes:

"The graphical features of the artists' books and visual poetry are not anomalous. Their apparently exaggerated or idiosyncratic character simply calls attention to features of textual activity that are actually at work in all and any graphical artifact. If we attend to the way we model the representations of these works for digital presentation, then we can extract from them a critical vocabulary for articulating the way books work as aesthetic instruments."²

In this response I will seek to amplify and further problematize the ramifications of her theoretical approach for practical implementation. This is a reflection of my own personal interest in writing software for creating and displaying complex texts, as well as an acknowledgement of the fact that my time spent studying philosophy was focused on the quaint German tradition of Hermeneutics, leaving me with a dearth of brain cells able to absorb the added onslaught of French and American Structuralist theory. I will leave the debate about the intricacies of that portion of her argument to those in this audience who I have seen, over the last few days, eat such stuff for breakfast.

¹ Drucker, Johanna. "Visual / Digital / Poetical", Textologies Symposium, McMasters University, Hamilton, ON October 2004. p. 62.

² Drucker, p. 67.

Prof. Drucker's paper raises three main issues for me. The first is the challenge of instantiating non-deterministic action into deterministic computational systems. This can be generalized to an issue of embedding qualitative theory, of the sort we're discussing here in this workshop, into instruments. The second is the need to bring the creators of computer languages and operating systems further forward in the heteroglossic field. And the third issue is related to the question of how—and where, and when—do we intervene into the construction of these instruments.

Theory Into Action

"A basic question of my project," Prof. Drucker writes,

“...is [what] is it that we want to model in creating a data structure that can represent the reading provided by an aesthetic artifact? How can this be produced in a digital environment where it can be stored, analyzed, displayed?”³

Creating such a model poses theoretical, implementational and representational challenges. How do we formulate a model of reading? How do we codify that model into code? And how does the code get re-presented to the user?

At the Typography and Visual Communication conference in Thessaloniki last July, we were presented with four different models that sought to address the relatively straightforward question of how peoples' eyes behave when they scan text.⁴ That is four different models at just the physical layer, an embodied activity that is amenable to empirical tests. When we move further into the act of reading and start considering the physiological aspects, and then yet further to consider the phenomenological aspects, we see that converging on an agreeable model of reading becomes exponentially more complicated.

If we stay at the phenomenological level, this poses no great problem. We can argue back and forth about whether we should be employing a structuralist approach, or a hermeneutic approach, or some form of speech-act theory. But when we turn to the second question—how to codify that model into an application which is instantiated via a programming language—things get messy.

For all their seeming variety, mature computational machinery all follows the same basic procedure: Boolean logic manipulating gates on a circuit, where the logic must be processed in a hierarchical, sequential and ordered manner. We can build whatever complex, multivalent, polymorphous scaffolding we want on top of this foundation, but at the point where the rubber hits the road—the point at which something *happens*—the scaffolding collapses to a linear sequence. For this reason, while I applaud Prof. Drucker

³ Drucker, p. 77.

⁴ Larson, Kevin, “The psychology of word recognition”; Enneson, Peter, “Visual interference, response bias, computation costs and cue value...”; Lund, Olav, “Putting legibility research in its right place: a history of legibility research”, Proceedings of the 2nd Annual Conference on Typography and Visual Communication, Thessalonki, Greece, July 2004.

interest in investigating ways of representing digital media that partake of a quantum order, until we have a fundamental shift in the construction of computing machinery—that is, until we have a viable quantum or polylogical computer—I am skeptical of our software’s ability to escape its fundamental, deterministic linearity.

In a way, though, a critique at this level obscures a more fundamental question. That question is, why do we *want* to provide people with the *experience* of reading such texts? Why do we want to attempt to recreate a phenomenological experience through instrumental means? I understand the frustration—I share it with my own work—of not having an appropriate schema for re-presenting the full linguistic and visual content of something like Prof. Drucker’s *History of the/my Wor(l)d*. Within the computational framework, such works get flattened; some things get lost; important aspects are made invisible. However, may it not be the case that, if we are really committed to not doing violence to the work, we should back away from an attempt to re-present the reading of the work? Should we not leave that to, well, the *actual reading* of the work?

Artificial Language Heteroglossia

Suppose that we answer that question in the negative. Then our efforts at capturing the essential qualities of complex text works require that we embed a theoretical model into an instrumental form. The act of embedding, I would argue, starts before the writing of whichever scripting scheme—XML, TEI, etc.—we choose to help us do so. All of these schemes require instantiation within a high-level language—C/C++, Java, Visual Basic, etc. Each of these languages come loaded with all sorts of assumptions about what constitutes data, how data should be manipulated, and what is valid output from that data. These languages are designed, often by as few as one person and rarely more than a handful of people. The agendas of these individuals carry forward into the languages that they create. Let me read a passage by Larry Wall, the designer of the Perl scripting language, from a talk that has become famous in popular computing circles called “Perl: the Postmodern Computer Language”:

“How does Perl put the focus onto the creativity of the programmer? Very simple. Perl is humble. It doesn't try to tell the programmer how to program. It lets the programmer decide what rules today, and what sucks. It doesn't have any theoretical axes to grind. And where it has theoretical axes, it doesn't grind them. Perl doesn't have any agenda at all, other than to be maximally useful to the maximal number of people. To be the duct tape of the Internet, and of everything else”.⁵ (Wall)

If we are really committed to respecting the heteroglossia of work such as that with which Prof. Drucker is concerned, we need to bring actors such as Mr. Wall into greater focus. If they have done their job well, they have provided us with an extremely rich vocabulary. But they must, inevitably, choose to include some computational utterances and exclude others, Mr. Wall’s claims of apoliticality notwithstanding. In short, the language designer determines *what* can be said. It is perhaps only because of ignorance, but I do not know of any scholarship that has seriously looked at the origins of these

⁵ Wall, Larry, “Perl: the postmodern programming language”, <http://www.wall.org/~larry/pm.html>.

programming languages with an eye towards the ontological ramifications of their design. Only by doing so can we situate them correctly in the heteroglossic field, and can we understand their *material* consequences.

This is an appropriate place at which to mark an objection I am sure is waiting in the wings, namely, why stop at high-level programming languages? Is it not true they get translated into low-level languages such as assembly, which then gets translated into machine code, which then gets translated into binary, which then gets translated into circuits, all of which are the result of human design? Or, as the little old lady from Hawkings *A Brief History of Time* would have it, isn't it turtles all the way down?⁶

Unfortunately, it *is* turtles all the way down. But that does not much bother me, as I am not interested in finding and articulating the Prime Designer. I *am* interested in problematizing all of these layers as a theoretical project; as a practical project, I *am* interested in focusing on the high-level languages because that is an area in which we actually have a reasonable amount of choice.

Generalization of problem to scene-description/semantic web

Regardless of whether we can satisfactorily resolve these two issues—identifying the goal of mark-up schemes and locating the role of language designers—we must have ways of cataloguing, archiving and disseminating information about artists books, visual poetry and other complex textual artifacts that involve an apprehensive engagement that falls somewhere between reading and looking. Thus, given that these efforts will continue, I am intrigued by the ways in which they can influence work on other schemas for describing content. An example is the definition of the MPEG-7 standard, which is an attempt to define a standardized method for describing multimedia content.⁷ Part of what is interesting about the MPEG-7 effort is that content is considered a secondary characteristic of the data, and is thus referred to as “metadata”. Here we see an interesting perversion of the way in which content is seen at least from the fine arts perspective, where at the very least content and media are considered to be co-equal and co-dependent. This perversion arises out of decisions buried deep within the history of the development of digital media, including the transformation of all content into equivalent bits in order to facilitate its orderly transmission through the circuits and between the machines. And this is precisely the reason why more people from the humanities world must get involved in the formulation of these standards, where they can bring into the process their concerns related to technology embedding a particular world-view. Another domain where this could prove fruitful is that of the Semantic Web, which is the effort to make the web meaning-rich and not just data-rich.⁸ The people developing the Semantic Web talk easily of ontologies, but it is an impoverished conversation that makes no effort to critique or problematize the ramifications of *different* ontologies.

⁶ Hawkings, Stephen W., *A Brief History of Time*, Bantam, New York:1998.

⁷ Martínez. José M., ed. “MPEG-7 overview”, <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

⁸ Berners-Lee, Tim, James Hendler and Ora Lassila,, “The Semantic Web”, *Scientific American*, May 2001.

So, those are my three questions, and they are for myself as much as for Prof. Drucker: why do we want to re-present the phenomenological act of reading; how do we understand the “co-authorship” of such representations by programming language designers; and how do we—as humanists, as aestheticians, as critical theorists—participate in the definition of the textologies which are being developed at this very moment?

Let me close by taking a moment to suggest a connection between Prof. Drucker’s paper and that of the others we have discussed so far. The desire that leads us in the attempt to codify a mark-up scheme—or any digitally embedded ontology—for artists books and visual poetry, is the same desire that leads us to encode more and more of our personal information into the technosphere, and thus become active accomplices in the building of Prof. Kroker’s biometric subject. It is the same desire that leads us to actively seek out Prof. Feenberg’s decontextualized no-space, and in so doing, subject ourselves and our creations to an ever greater degree of instrumentalized control. It is the same desire that leads some to revel in the efficiency and magnification promised by life on a purely technological plane, and from there to exchanging the humanity of Prof. Winner’s concern for greater ease in exchanging data. It would seem from these resonances that the desire for instrumentality, as it travels the path from necessity to convenience, is contagious.