P.o.E.M.M.: Bigger on the Inside

Darren Wershler
2015 03 26 v4

Anthony Burgess wrote more than once that a book could be imagined as its backronym: a Box of Organized Knowledge[1]. A P.o.E.M.M.—Poem for Excitable [Mobile] Media—is no different in the sense that some kind of linguistic compression is at work in it as well.

But P.o.E.M.M.s exist on the other side of the Gorilla Glass window that divides digital text from print. Because they are software, P.o.E.M.M.s and other apps also involve the bundling and compression of computer code. And, in an age when books no longer hold the monopoly on dense media experiences, apps offer a powerful new kind of expansion. The animations that accompany a contemporary program's activation, like Apple's "Genie" and "Scale" effects, are metonymies that hint at what is to follow.

Before you even touch the screen of your phone, you expect your apps to be bigger on the inside. William Gibson imagined that software would produce this sense of anticipation when we opened it, and soldered his description of the user experience into a gleaming alliterative line: And flowed, flowered for him, fluid neon origami trick[2]. Plain prose was not enough to convey it. In Gibson's novels, the appearance of poetry is an index for an intimate encounter with technological complexity. In Jason Edward Lewis and Bruno Nadeau's P.o.E.M.M.s, poetry and digital media reflect each other in a bid to lure us down glossy, apparently infinite corridors lined with language.

With contemporary software applications for tablets and phones, the compression of medium, platform, format and genre can make it difficult to discern what, exactly, lies before us. For Apple, in particular, this is intentional: the fantasy of a "Seamless Experience" attempts to sharply differentiate our digital interactions from the days of giant putty-coloured cases, dangling wires, exposed circuit boards and the endless shuffling of removable media. As in Gibson's fictional world, the point for a corporation set on bedazzling consumers is to make the hardware appear to disappear.

So right now we know more about what the app expects from us than what it actually is.

The age of incunabula—the first printed books—lasted about fifty years, from the 1450s to around 1500. Volumes of printed lyrics were around for quite a while, then, before people knew how to talk about them or what to do with them as there was nothing to compare them to, nor printed history to document them. Following on the work of Matthew Fuller and the other early practitioners of software studies, Anders Fagerjord poses the question: Do we need an app studies?[3]

Probably. Fuller convincingly argues that software studies should "show the stuff of software in some of the many ways that it exists, in which it is experienced and thought through, and to show, by the interplay of concrete examples and multiple kinds of accounts, the conditions of possibility that software establishes"[4]. He goes on to assert that Software Studies also needs to confront the "kinds of thinking and areas of work that have not historically "owned" software, or indeed often had much of use to say about it"[5]. Thus, there are many factors to consider:

> algorithms; logical functions so fundamental that they may be imperceptible to most users; ways of thinking and doing that leak out of the domain of logic and into everyday life; the judgments of value and aesthetics that are built into computing; programming's own subcultures and its implicit or explicit politics; or the tightly formulated building blocks working to make, name, multiply, control, and interrelate reality.[6]

It is also worth noting that this list, from a book published in 2008, is too old to contain the word "app."

"App" is short for application. Born of mobile media, the app is a relatively recent addition to the software menagerie. The iPhone first appeared in 2007, and the App Store followed in 2008. Apps are usually purpose-built software with a handful of specialized uses rather than general applications, and are usually designed for mobile hardware.[7] They do bear a family resemblance to the now-nearly-moribund category of desktop "widgets," but many widgets always felt like an attempt to retrofit desktop software to look and behave like apps. Fagerjord's definition is useful because it is agnostic about what apps actually contain; what matters is what they do. An app "orchestrates the networks, input/output and sensors for a purpose the user finds useful or entertaining, or both. An app can make calculations, based on input from the user or the sensors, send and receive data over a network, and output the results to the user, and simultaneously send the results over a

network".[8] The key idea here is the *orchestration of networks*. Apps are nothing if not a complex set of relations between people, technologies, discourses, objects, companies and other institutions.

As Jonathan Sterne has argued so eloquently in his work on the mp3, software application formats are also material. A format "is a crystallized set of social and material relations. It is an item that 'works for' and is 'worked on' by a host of people, ideologies, technologies and other social and material elements."[9] Sterne's language invokes that of Marx in his first volume of *Capital*, where he describes a commodity as a crystallized form of the social substance of labour.[10] Clearly, there is a politics at work here, relating to the ways in which apps, like all objects, are produced by the exploitation of human labour.

Another way of saying that apps are sets of relations is that they are imbricated within particular social and technological networks. As apps circulate through these networks, they transfigure the people, institutions and discourses that they touch. As a result of such contact, apps change, too. Thus, if we want to understand what apps are, we need to begin documenting their entire cycles of production, circulation and consumption.

Some major differences between apps and more familiar types of software are their economics as well as technologies. Apps are tied very closely to specific hardware platforms *and* networks of distribution[11]: an app is an application distributed through an integrated, monopolistic outlet[12]. For instance, an iPhone app can only be created with Xcode, Apple's developer software. Similarly, testing the app on an actual iPhone requires a developer license from Apple, which the developer must pay for on an annual basis. A normal license permits testing of the app on up to five iPhones. However, the final app must be distributed through the App store and go through a mandatory review process before being made public. The developer can then choose to distribute the app for free. However, if they choose to sell it, Apple takes a cut.

This sort of vertical integration has characterized both print and digital publishing for several decades. Moreover, it is an ongoing process that shows no signs of abating any time soon. Rhetoric about its effects tends to be polarized between neoliberal paeans to "disruptive technology" on one side and laments about the decline of the public sphere on the other. From the eye of the hurricane, it is difficult to assess how this process will resolve; however Lewis and Nadeau's experiments with the app format constitute an intervention into assumptions held around apps. What should an app cost? Does

electronic art function differently when it is projected onto the gallery wall than it does when it is in in our pockets? What would it mean to take a P.o.E.M.M. out of circulation? Where would you store its limited editions? The P.o.E.M.M.s inevitably pose these sorts of questions while rejecting the delivery of past answers. Lewis and Nadeau have no truck with nostalgia, nor do they rely on familiar aesthetic codes (such as the pixelated, blocky appearance of many "indie" games and programs, which often function as a sort of team uniform for digital art that makes critical claims) to gesture toward their politics. Lewis has observed in conversation that he is always a little bothered by discussions of the economics of app publishing that only highlight the constraints while ignoring the benefits: worldwide distribution; guaranteed execution on a unique combination of operating systems and first-rate hardware and OS; and significant and focused developer support for coding. At the same time, he recognizes that Apple and other such gatekeepers can provide real resistance and even censorship in the face of projects that present political, formal and aesthetic challenges. The ambivalence is real, and, since we are thrown into the middle of the experience, we have no choice but to engage it from our necessarily limited perspective.

Ambivalence is intrinsic to this project. As Lewis explains, the P.o.E.M.M.s have their relative beginning in documents from a 1964 Louisiana court case that was attempting to racially classify an adopted child. In the records, the judge claimed that the maintenance of the general public's welfare actually depended on such determinations. Lewis writes, "That claim seems cartoonishly hyberbolic, until I remind myself that I was adopted only three years afterwards, in 1967, a Cherokee/Hawaiian/Samoan boy given a loving home by a White family from rural northern California." For Lewis, the judge's claim about racial classification lays bare the ways in which exclusionary principles in such official discourse, no matter how cartoonish they may seem in isolation, have a central role in the managerial regime of contemporary society. Even as we try and disown or disqualify them, they continue to position us. One possible tactic in the face of this knowledge is, precisely, formal: to attempt to find ways of being and making that are cognizant of their own historical limits, yet nevertheless attempt to transgress those limits by returning to them from subtly different perspectives.

And there is something new emerging here, as the name of the series claims: a space in which conflicted identities try on ambivalent forms, to varying degrees of success. Each P.o.E.M.M. is a proposition, a possible genre with its own conventions, waiting

to be articulated to one of several possible politics. On first release, a P.o.E.M.M. is *sui generis*, a field of one, necessarily protected in order to say difficult things. An exhibition version of the work (not yet mobile, thus the brackets around '[Mobile]') precedes or appears simultaneously with its release as an app. Each P.o.E.M.M. follows a cycle that theoretically allows its specific emergent form to explore different modes of authorship, collaboration and distribution. A gradual opening up of the work would mimic the cycle that many pieces of software have now followed, from "scratching a developer's personal itch" (as Eric Raymond, paraphrasing Linus Torvalds, describes it[13]) to making a formal contribution to the open base of computer code that powered the meteoric growth of digital networked culture. This is one of the ways in which the P.o.E.M.M. as genre draws its cues from formats developed in the tech industry rather than literature or art, which follow this trajectory toward openness much more slowly. The subsequent opening of the app to other writers, artists and coders transforms the work into a genre; where five poets were invited to write into the app, thus, making it an anthology. Next, the expansion of the app code allowed users to input text into a Twitter feed, whereby the app becomes a tool for composition and a vehicle for sharing new compositions with others. In effect, the P.o.E.M.M. becomes a platform. In the final stage, the code is released under a public license of some sort, allowing for the cycle to begin again in the hands of others.

The language of potential in the preceding paragraph is deliberate. As of this writing, none of the eight existing P.o.E.M.M.s have gone through the full cycle that Lewis describes. *Speak* has seen the most iteration, as it has been developed to the platform stage; *Know* currently exists as an anthology. Part of the problem is the availability of iOS programmers within the academy, where such skills are still relatively rare among graduate students, nevermind university faculty. Lewis hopes to locate a new programmer shortly in order to iterate *Speak* (at least) through its full cycle.

Why bother with such an elaborate process, given the difficulties involved? The larger vision for the P.o.E.M.M.s—beyond the already significant fact of their very existence—represents a practice of utopian coding. For Fredric Jameson, our contemporary "constitutional inability to imagine Utopia itself" has nothing to do with the failure of individual acts of creativity. Rather, it is the "result of the systemic, cultural, and ideological closure of which we are all in one way or another prisoners."[14] In other words, the precise point at which such an ambitious practice fails represents the current limits of what we will allow ourselves to know, do and think.

The proposed lifecycle of a P.o.E.M.M. reflects a mundane set of editorial decisions like those that surround any publication, yet it also draws attention to the economics and strictures of writing in a digital milieu. Some of these strictures are aesthetic and have to do with things like the possibilities the P.o.E.M.M.s present for rethinking long-standing aspects of poetic form like lineation. This is not to say that digital poetics begins with apps. Digital poetry has been developing "new" conventions for poetic lines for decades, and many of these are quite stable (see, for example, Brian Kim Stefans's "Dream Life of Letters," an early attempt to exhaust the formal vocabulary of the electronic poem[15]). However, these conventions have not yet been cataloged and described in a systematic manner. From a literary perspective, digital poetry requires someone to write something that might be a cross between Apple's famed *Human Interface Guidelines* and Dennis Cooley's "Breaking and Entering" essay[16]. However, this is the easy part of the work that remains to be done.

The more difficult part has to do with discerning the political economy of writing and publishing in a networked digital milieu. Beyond what we can see on our home screens or in the app store, there are behemoths making the rules. However, they only become visible when you act unexpectedly. (Being of a certain age, this always makes me think of The Silver Surfer hitting the invisible barrier surrounding the Earth when he attempts to leave through apparently open skies—as good a metaphor for ideology as any). When confronted with the reality of the walled garden that the Internet has become, one possible response is to read the situation as an apocalyptic end to innocence: the end of the Web as a front-end for applications, and thus the end of open standards. Yet, software was never innocent, as Fuller reminds us: much software comprises simply and grimly of a social relation made systematic and unalterable. (Consider, for instance, the ultimately arbitrary informational regimes governing who is inside or outside of a national population)."[17] In a similar vein, Lewis observes that "a software application such as Word, for instance, can be thought of as an orderly and (mostly) predictable assemblage of biases that reify the imagination of its creators into executable code."[18] One of the virtues of P.o.E.M.M.S is not that they behave otherwise, but that they can *imagine* behaving otherwise.

This is the virtue of a utopian approach: P.o.E.M.M.S are bigger on the inside. As a format, they offer infinite room for other inhabitants because they deal with possibility and are

unafraid of failure. In various stages of their development, different people can reach into them after the fact and change what they find. This is the private utopia of coding: individuals or small groups of people finding a modicum of time and resources, often in the pockets of larger hierarchically ordered organizations, while ostensibly at work on something more important. Fuller calls this phenomenon the "self- sufficiency" of software,[19] because it allows for a kind of distance from social and cultural norms on one level even as they determine our situation on another.

There is a spirit of generosity about these works that I find very appealing. Poems have always been an oddity as a medium of exchange because their value is totally arbitrary—they mean everything to some and nothing to others. Literary value hangs, in part, on questions of format: first editions, signed and numbered multiples and rare books are (usually but not always) worth more than poems scrawled on a wall with spray paint. Lewis and Nadeau have found a way to take the after-purchase modification of a digital work, which is usually a force for evil (cf. the nerfing of PVRs to remove their ability to skip commercials, the loss of a treasured feature in a favourite program, the introduction of promoted content in a paid app that previously had none, the steady bloat of once-reliable applications, et cetera.) and turn it into yet another unexplored artistic technique. Imagine Margaret Atwood in a black jumpsuit, sneaking into your house after dark to erase some sentences in your copy of one of her books, only to write new ones in their place.

Another aspect of this generosity on the part of the programmer-writers is the abrogation of mastery. Others might be better at the form that you invented, and what of it? For Fuller, software is "a space for profound and unfinishable imagination."[20] Beyond software, the history of media technology is a history of people discovering better uses for things after the fact. Allowing others inside is a way of making the naturalized, shiny surface of the app into a first step toward grappling with what an app is. Open code (or the current canard of teaching everyone to code) is not the only answer; we need to be able to account for what people do with things in various stages of their creation, circulation, consumption and destruction. What we need, then, to make sense of all aspects of the P.o.E.M.M.s and the apps that follow them, is some sort of continual process of elaboration—something like Latour's processual circle in *The Politics of Nature*[21]. What lies ahead, for better and worse, is a point where technology and art become indistinguishable.

1 Burgess, Anthony. "*Confessions of the Hack Trade.*" The Guardian. 4 March 2012. First published in The Observer, 1992. Accessed March 26, 2015. http://www.theguardian.com/culture/2012/mar/04/anthony-burgess-on-journalism-1992

2 Gibson, William. *Neuromancer*. New York: Ace Books, 1984. 52.

3 Fagerjord, Anders. "*Toward App Studies.*" Presented at Internet Research 13.0, University of Salford, 21 October 2012. http://fagerjord.no/blog/?page_id=95.

4 Fuller, Matthew. "*Introduction, the Stuff of Software.*" Software Studies: A Lexicon. Ed. Matthew Fuller. Cambridge/London: The MIT Press, 2008. 1-14. 1-2.

5 ibid., 2.

6 Ibid., 1.

7 Fagerjord 1.

8 ibid., 5.

9 Sterne, Jonathan. "*The MP3 as Cultural Artifact.*" New Media & Society 8.5 (2008): 825-42. 826.

10 Marx, Karl. *Capital: A Critique of Political Economy. vol. 1.*: Penguin Classics. 3 vols. London/New York: Penguin Books in association with New Left Review, 1990. 27, 29.

11 Fagerjord 2.

12 ibid., 4.

13 Raymond, Eric. "The Mail Must Get Through." Homesteading the Noosphere. The Cathedral and the Bazaar. version 3.0. 2000. http://www.catb.org/esr/writings/homesteading/cathedral-bazaar/ar01s02.html

14 Jameson, Fredric. "*Progress Versus Utopia; or, Can We Imagine the Future?*" Science-Fiction Studies 9.2 (July 1982): 147-158. 153.

15 Stefans, Brian Kim. "*The Dreamlife of Letters.*" Toronto, Coach House Books, 2000. Accessed March 26, 2015. http://archives.chbooks.com/online_books/dreamlife_of_letters/

16 Cooley, Dennis. *The Vernacular Muse: The Eye and Ear in Contemporary Literature*. Winnipeg: Turnstone Press, 1987.

17 Fuller 3.

18 Lewis, Jason Edward. "*Preparations for a Haunting: Note Towards an Indigenous Future Imaginary*". In The Participatory Condition. Eds. Darin Barney, Gabriella Coleman, Christine Ross, Jonathan Sterne and Tamar Tembeck. Minneapolis: University of Minnesota Press. Under revision; scheduled publication: Fall, 2015.

19 ibid., 6.

20 ibid., 7.

21 Latour, Bruno. *Politics of Nature: How to Bring the Sciences into Democracy*. Cambridge: Harvard University Press, 2004.