

The Aesthetics of Generative Literature: Lessons from a Digital Writing Workshop

DANIEL C. HOWE AND A. BRAXTON SODERMAN

Brown University

ABSTRACT

This paper explores a range of issues related to the pedagogy and practice of generative writing in programmable media. We begin with a brief description of the RiTa toolkit — a set of computational tools designed to facilitate the practice of generative writing. We then describe our experiences using these tools in a series of digital writing workshops at Brown University in 2007-2008. We discuss and theoretically examine a set of core issues raised by workshop participants — distributed authorship, the aesthetics of surprise, materiality, push-back, layering, and others — and attempt to situate them within the larger discourse of generative art and writing practice.

INTRODUCTION

'A poem is a small (or large) machine made out of words.' -William Carlos Williams

This paper explores a range of issues related to the pedagogy and practice of generative writing in programmable media. We begin by presenting a short description of RiTa ^[1], a set of computational tools developed by one of the authors to facilitate the practice of generative writing both within and beyond the classroom. While the RiTa toolkit provided the technical framework for the courses we discuss ^[2], this paper focuses less on the details of RiTa (discussed elsewhere ^[3]), and more on a recurring set of questions, concerns, and concepts that arose for students as they engaged with these tools to implement their own creative works. This set includes ideas of distributed authorship, programmers as authors, the aesthetics of surprise, the materiality and push-back of a medium, the layering of digital texts, thinking discretely, and the process of "roughing-up" texts. While it is clear that these ideas emerged within the specific context provided by the tools and pedagogical strategies of the class, it is our belief that they represent central questions for practitioners in the field and that the various affordances of the tools serve only to increase the frequency and amplitude of their signal. Our intent here is not to propose a rigorous definition for what is or isn't generative writing ^[4], but rather to present what may best be described as "lessons learned" from our first-hand experience with students as they engaged, often for the first time, with generative language systems. By reflecting critically on such concerns, it is our hope that we can both challenge and extend current thinking on generative art, and, more specifically, on generative writing itself.

RITA

The RiTa Toolkit for Generative Language is a suite of open-source components, tutorials, and examples that provide support for a range of tasks related to the practice of creative writing in programmable media. Designed both as a toolkit for practicing writers and as an end-to-end solution for digital writing courses, RiTa provides support for a range of computational tasks related to literary language including text analysis, generation, animation, display, text-to-speech, web-based text-mining, and interfaces to external resources (e.g., WordNet ^[5]). Students from a wide range of backgrounds (creative writers, digital artists, media theorists, linguists and programmers, etc.) have been able to rapidly achieve facility with the RiTa components and thus move quickly onto their own creative language projects. As RiTa is designed to support integration with the Processing environment for arts-oriented programming ^[6], students have immediate access to a large community of practicing digital artists, and can easily augment RiTa's functionality via the vast collection of libraries available.

RiTa was designed with several practical goals in mind, specifically a) to implement an end-to-end tools set for use across a variety of digital literature courses and workshops, b) to make available (both to students and practicing writers) new procedural techniques to enhance writerly creativity, c) to enable the development of resources to increase productivity across typical writing tasks d) to accommodate users with a wide range of backgrounds and levels of technical expertise, and e) to spur the creation of new literary forms. Further, we hoped that these tools would be applicable for users working in a variety of disciplines, not only within creative writing workshops, our focus here. Other potentially viable areas include general language education (English, ESL, etc.), natural language generation tasks, and even computer-augmented literary criticism. High-level design goals for the software included enabling a) experimentation with generative language systems without the large structural and cognitive overhead typical of such systems, b) simple distribution and sharing of prototypes, projects, and code via the web, c) the creation of new literary and artistic forms augmented by computational practices, and d) new insights into design principles for researchers interested in providing creativity support tools for work in natural language.

It has been our hope that RiTa will contribute to the field of creativity support by providing an integrated, end-to-end toolkit for a range of users, as well as helping to re-situate the production of literature among more traditional applications for computational art-practice (film-making, digital photography, video editing, music composition, etc.). Further, since the system's development process has closely followed recent research in design principles from the creative support tools (or CST ^[7]) community, we hope to provide some initial feedback on the efficacy of these principles through their application to the relatively new domain of literary language.

GENERATIVE ART AND LITERATURE

Before preceding, we should discuss briefly what we mean by the term *generative literature*. One recent definition can be found in Jean-Pierre Balpé's text "Principles and Processes of Generative Literature: Questions to Literature." He writes that generative literature is "the production of continuously changing literary texts by means of a dictionary, a set of rules, and/or the use of algorithms" ^[8]. This definition follows somewhat closely from *generative art*, of which generative literature may be considered a sub-genre, though

one with unique properties stemming from the textual medium. For example, Philip Galanter describes generative art as "any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to, or resulting in, a completed work of art" ^[9]. Marius Watz adds that such systems often "employ dynamic rather than deterministic processes, and are created by an artist but rarely completely under her control" ^[10]. There are many examples of this work ^[11], much of which, contrary to popular conception, predates the use of computers (with examples in Conceptual Art, Minimalism, Fluxus, OpArt, etc.). For the purposes of this paper, we will consider a somewhat broader definition than Balpe's above, including a continuum of language-based work (whether continuously changing or not) that employ generative processes as defined above. To help situate the reader, table 1 lists a variety of hypothetical works that might be included within the scope of generative literature.

Table 1. A Continuum of Generative Literary Examples

1.	An instruction set for manually creating a text: a) copy the contents of an article from the New York Times b) substitute all instances of 'they' with 'I' c) etc.
2.	A printed page of text labeled as 'an output from computer program x'
3.	100 printed pages labeled as '100 iterations of computer program x'
4.	A real-time program that display a new iteration of a text each time it is clicked
5.	A real-time program that displays (one-by-one) all possible permutations (a finite #) of a generated text
6.	A real-time program that continuously scrolls a text (adding a new sentence to the end every 2 seconds) without user interaction
7.	A real-time program that continuously mutates the content of a displayed text by substituting words and phrases from a dictionary
8.	The source code for a program that generates any of the above outputs
9.	The source code for a program that generates any of the above outputs

THE GENIUS VS. THE ENGINE

Beyond simply stating what generative literature is or what kinds of work it might include, it is useful to situate the concept in a broader historical and cultural context. Indeed, the word "generative" itself offers insight into the relationship between traditional forms of authorship and more recent computational writing practices. "Generative" is etymologically connected to words such as "genius" and "engine," both of which

stem from the notion of begetting or engendering. Early forms of the word engine were synonymous with connotations of genius (as natural talent, ingenuity, innate ability) though today this use of engine is archaic while its use as "machine" or "tool" is common. The splitting of the idea of begetting (into the different meanings of the words engine and genius) created divergent paths that eventually emerge in the form of a confrontation between the innovative powers of man and the rumbling efficiency of the machine: on the one hand, genius as a form of natural and almost mystical inspiration, the epitome of the human subject as creator, and on the other, the engine as an artificial generator of force, the mechanical object as creative producer. Such a confrontation finds recent expression in the pitting of chess engines against chess masters culminating in the 1997 defeat of Garry Kasparov by IBM's "Deep Blue" chess engine. Indeed, it seems that in the last half of the 20th century we have witnessed the waning of genius and the waxing of the engine that now populates the digital landscape as search engines, game engines, physics engines, poetry engines, etc. Of course, the notion of the genius or inspired originator is far from effaced, especially in the realm of digital literature. Some theorists look forward to the coming of digital, literary masterpieces, while others argue that genius is now encoded into algorithms, where the ingenuity and intention of the programmer (as author) finds its home.

It is well known that much of the history of digital media and computation has sprung from a desire to preserve human creative "genius" when confronted with the growing mathematical and informatic complexity of the world. Thomas Edison once quipped, "Genius is one percent inspiration and ninety-nine percent perspiration," yet advancements in digital production techniques have perhaps sought to reverse this ratio. In fact, many of the seminal thinkers in the history of digital media – Vannevar Bush, J.C.R. Licklider, Douglas Engelbart, Ted Nelson, etc. – were motivated by a desire to automate arduous tasks (calculating equations, organizing, sorting and retrieving information, etc.) so as to free the creative intellect from rudimentary, repetitive activities. Even L. F. Menabrea, discussing the benefits of Babbage's Analytic Engine, wrote in 1842: "what discouragement does the perspective of a long and arid computation cast into the mind of a man of genius, who demands time exclusively for meditation, and who beholds it snatched from him by the material routine of operations!" ^[12] A student in a recent workshop made a similar point (though without the invocation of "genius"):

If I had the time and patience, I could do it by hand. Software is my executor, it does what I would do if I was there. Of course, even if such time and patience actually existed, I still would not do it by hand. What would be the point?... It is far more interesting to leverage what the machine is good for, endlessly exacting repetitive tasks, to achieve our creative goals.

Yet, as much as innovations in computation have been envisioned as preserving genius or facilitating human creativity, it is undeniable – especially within the realm of art – that the engine can be used to challenge notions of the inspired genius, especially where generative systems assert their own autonomy in the creation of the artwork. Jim Carpenter writes,

...the Poetry Machine that realizes these propositions, words in meaningful combinations, originality, and cohesion, will have nudged computed poetry out of mere novelty and passing diversion. It will have composed texts worth reading. And most importantly, it will have obsolesced the Author and rendered Him irrelevant ^[13].

In this statement we find the poetry engine displacing the author entirely. Yet, arguing for one pole over another – genius over engine, engine over genius – is less interesting than their synthesis. Our use of the word "generative" is intended to mark this synthesis, to mark the collaboration of the human creator with the machine creator, the cybernetic feedback loop of digital authorship. In fact, as we discuss below, the most successful student works using RiTa were not those where the generated text was "random" or freely created by the machine, nor those composed "offline" by a human author and later transferred into digital media, but those where aspects of the resulting text were discernible as co-authored by both artist and machine.

FOCUSING ATTENTION ON THE ENGINE

What do teachers of digital literature gain by viewing pedagogical tools in terms of the 'engine'? First of all, it is commonly asserted that genius is un-teachable, that individuals are "born with genius," that it is "in their nature." On the other hand one might claim that the engine is precisely what *is* teachable; or more broadly put, that the engine is a tool or technical system that one can be taught to use effectively. If traditional writing workshops still implicitly harbor the residue of the author as "genius" – with students aspiring to write the 'great American novel' or to become immortalized like the great modernists often read as examples – then recognizing the work of the engine can temper lingering ideas of individual genius (and its associated concepts) with the more collective intelligence of the engine. The various "engines" that the students used to drive their artworks (the algorithms, data sources, and technical apparatus "under the hood") were often composed of multiple support libraries and APIs in addition to code extracted from others (often found on the web) and code they had written themselves. This tapestry of code produced a network of "distributed authorship" (often including "invisible" partners) which served to further challenge the concept of individual authorship ^[14]. Indeed, successful projects often emerged not only from the ingenuity of the artists themselves, but from the combination of algorithms and libraries assembled from other (acknowledged or unacknowledged) participants. One might even conceive of such distributed authorship as an emergent phenomenon, with complex behavior arising from the integration and collective activity of many simple parts. In our experience, foregrounding this network of distributed, often invisible, authorship helped to alleviate the burden of individual genius and the "anxiety of influence" (i.e. students worrying about creating something completely new or unique, or asking themselves "Do I have the 'gift' of writing?"). Students were able to build on the work of others and began to see the creation of artworks as a collective effort. Clearly, generative literary works are often (if not inevitably) a form of collaboration – between the writer and the programmed system, and further, between the multiple "authors" of said system. This insight, rather than discouraging students, aided in creating an open environment where ideas (and code) circulated freely. Moreover, focusing on the multiplicity of authors and engines that construct such works aided

students in understanding what many have identified as an essential attribute of generative art production: the relinquishing, at least in part, of the artist's control over production (certainly a common trope in contemporary art of the last century).

Finally, foregrounding the engine allows us to differentiate our pedagogical approach from those who tend to view the effects of the digital on writing primarily in terms of distribution; whether through blogs, animated texts, or 'Web 2.0' services, etc. Such approaches – see the WIDE Research Center Collective as one example – tend to focus on the surface level of the text, often ignoring the potential of the machine as a powerful means of processing, analyzing and generating literary artifacts ^[15]. Highlighting the "engine" as an important element of digitally mediated writing counters the common tendency to black-box the technological and its potentially vast affect on the practice of writing ^[16].

GENERATIVE PEDAGOGY

While providing appropriate tools for the production of digital literature is perhaps a necessary condition for an understanding of how writing changes when performed in computational media, it is clearly less than sufficient. An accompanying pedagogy that references the specific tools and procedural strategies of the medium is equally important. Further, it can be highly productive for both student and instructor when these two elements are tightly coupled and developed in a mutually-informing fashion. Such a coupling implies at least communication, if not close collaboration, between those creating the tools and those developing the accompanying intellectual program. In this regard we were in the fortunate (though perhaps rare) position of having some control over the ongoing development of both the tools in question and the accompanying pedagogical material (readings, assignments, discussions, critiques, etc). In several cases, specific materials were chosen to reflect important aspects of the technology being used. Perhaps more unusually – and more interestingly – were cases of the converse, where software tools were modified and/or extended in response to intellectual concerns raised in discussion. Though the practicality of this situation may be questionable at larger scales this should not prevent us from taking note of its benefits.

It is only in this context, where both access and inclination to tinker ^[17] have been so effectively limited by a range of economic and ideological agendas that such practices have ceased or slowed (i.e. copyright laws and digital rights management). The mechanisms and dynamics of such "limiting" is a topic that deserves closer attention, yet for our purposes it is perhaps enough to note that experiencing the mutability of technology in the classroom can be a step toward a) liberating students from artificial limits placed on their use of, and relationship to, technology, and b) in training future educators whom we might hope to expand on this practice in the future. In fact, a distinctly "generative" pedagogical approach was developed over the span of courses at Brown that, interestingly, aligns rather closely with recent teaching strategies for digital media ^[18]. Although beyond the scope of this paper, we present a brief comparison of this approach in Table 2.

Table 2. Traditional vs. Generative Pedagogy

	Traditional	Generative

Perspective	Top-Down	Mix of Bottom-Up & Top-Down
Model	Banking model: Teacher makes 'deposits' into student brains; focus on lectures, note-taking, and recitation (exams)	Everyone learns, everyone teaches (including the instructor). Students learn (in-depth) about subjects of interest, then present to each
Practice	Students work alone (guard against 'cheating'), focus on individual problem solving, memorization, recital of facts	Individual and collaborative projects with a range of group sizes; support reuse of 'found' materials and existing 'solutions', focus on finding/using resources in local and global communities
Evaluation	Fixed assignments with 'objective' grading by teacher, clear metrics & rewards	A mix of specific assignments and open projects (leveraging constraints) - evaluate via group critique. Consider context: students' backgrounds, skill-set, etc.
Goals	Satisfy fixed learning objectives, the same for each student, and for each iteration of a course	To learn how to learn and make 'personal' new knowledge - support different learning styles, goals vary from student to student
Structure	Hierarchical, based in institutional authority, knowledge is passed down from expert (teacher) to students	Networked, 'distributed authority', based upon shared-purpose, diversity of skills / backgrounds/ perspectives, sharing of resources, knowledge moves in all directions
Metaphor	Deterministic algorithm for which the output can be judged correct or incorrect (surprises are undesirable)	Generative / non-deterministic algorithm; outputs are judged subjectively (surprising outcomes are sought/valued)
Paradigm	Offline / Composed	Real-time / Improvisatory
Orientation	Modernist, focus on 'genius'	Post-modern, synthesis of 'genius' & 'engine'

THINKING DISCRETELY

Browsing the large database of generative works from the website Generative.net, or examining exhibitions such as Autopilot (2004), Generator.x (2005), and Generator.x 2.0 (2008), one is struck by the fact that the vast majority of the artworks focus on sound and image^[19]. Generative works that approach language from a computational perspective are, for the most part, eerily absent. There is, of course, a long history of generative literature, dating back into antiquity and extending into modernity as Florian Cramer has

excellently documented ^[20]. In 20th century art practice, for example, one can cite Tristan Tzara, Brion Gysin, William Burroughs, Raymond Queneau (and others from the Oulipo movement), Alain Robbe-Grillet, Jackson Mac Low, Charles Hartman, Florian Cramer, John Cayley, Eugenio Tisselli, to name just a few. Of course, the history of visual and sound-based aesthetic generation similarly claim lineages that date far into the past. Yet, given the strong history of literary experimentation the question becomes why there appears to be such a dearth of contemporary generative writing.

In order to answer this question one might begin with somewhat facile observations. For example, on the heels of the development of photography, phonography, and cinematography, the 20th century has been commonly perceived as a period where audio-visual cultural has expanded enormously. It is common to find research demonstrating how people spend less time reading literature today than watching films, listening to music, or playing videogames. Or, one might suggest that avant-garde aesthetic practice has migrated to audio-visual experimentation whereas literature, much like theater, is seen as antiquated and conservative. Further, one might claim that the growth of computational literary practice has been frustrated by the need for large (and traditionally difficult to obtain) corpora to facilitate interesting, complex text manipulation. It is only recently that individuals have had access to large/free databases of text (WordNet, RSS, Project Gutenberg, HTML pages, etc) to be used as raw materials in generative processes.

Another possible explanation for the relative lack of text generation in contemporary practice concerns the manner in which text can be decomposed into discrete parts. To follow this line, let us turn briefly to a comment made by a student in one of the courses taught at Brown University:

One advantage that text has over visuals is that because it can be broken down into atomic units (letters), you can create algorithms that produce alternative forms of text and create procedural and generative works. It would be difficult to accomplish such tasks with images, although if you were to use simple primitives, you can perhaps achieve similar effects. However, I think that with text, your results will be more consistent and meaningful, as opposed to the fractal-like appearance you might accomplish if you were to apply the same principles to visuals ^[21].

Essentially this student contrasts language — a discrete system of phonemes, morphemes, letters, words, etc. — with continuous, analog representations (images), that are harder to break apart into discrete units and thus harder to manipulate in meaningful ways. Moreover there is an assumption that these discrete lexical units can be thus manipulated due to conventional grammatical structures that may be emulated algorithmically, whereas no such structures exist for visual representations; no explicit rules exist specifying how parts of a particular image necessarily combine ^[22].

Interestingly, this line of reasoning would seem to predict a relatively higher frequency of linguistic generation as opposed to image-based generation. Yet, in the quote above, the student appears to consider only representational and figurative images, not abstract visualizations. While analog photographic images are not easily articulated into a system of smaller parts, they are, when digitized, articulated as discrete, atomic units; specifically pixels. At a fundamental level no "grammar" (analogous to the grammar of natural

language) exists to structure how pixels can be combined. Thus generative visual artworks tend to be abstract, or, as in the student's words above, given to a "fractal-like appearance." These works are often experiments in form without conventional signification, or, at minimum, they contain an abstract message loosely based on the "grammars" of nature or abstract art. Marius Watz has pointed out that "generative art is rarely concerned with figurative representation. In the few instances where the figurative is featured, it is usually in the form of raw materials (photography or video) for procedural re-interpretation" [23]. If generative artists working with digital images and sound use "a system of rules" to generate forms, these artists appear freer to experiment with these rules (because the system of pixels does not have conventional rules of how they should combine [24]).

In contrast, literary structures, though based on "discrete" levels of articulation, are already subject to strong rules determining how these atomic units can be meaningfully combined. This is not to say that literary artists cannot work on the level of the pixel; they can certainly do (and libraries such as NextText for Processing are available to facilitate such manipulation [25]). But, in such cases it is generally the appearance, style, or motion of the text that is altered, rather than the linguistic significance of the words themselves. One can also, in parallel fashion, treat the discrete units of language (letters, phonemes, words) as "pixels" in the sense that one ignores conventional grammar and sense in order to perform text manipulations on the atomic units of language: the *Nam Shub* text-processor by Jörg Piringner [26] would serve as one example of such an approach. The insight here, and one that is quickly appreciated by students as they begin to work in programmable media, is the importance of examining the various levels of articulation at play in the medium at hand, of working from a variety of vantage points and shifting focus from the big picture to the smallest elements. Applying this idea in teaching generative practice involves encouraging students to "see discretely," to decompose media elements into atomic, recombinant units capable of inspiring new generation strategies and aesthetic forms [27].

THE MEDIUM PUSHES BACK

Traditional, analog writing workshops – be they concerned with prose or poetry, traditional storytelling or experimental practice – generally avoid questions of technological mediation, though it is clear that even contemporary "analog" writers depend heavily on (and are shaped by) their use of computer tools [28]. On the other hand, digital writing workshops (particularly those utilizing tools like RiTa that bring the student quickly into contact with multiple technical layers) tend to foreground questions of technological mediation by situating the practice of writing within the broader framework of programmable media systems. Indeed, an important lesson from the classes taught at Brown was that engagement with technically-mediated practices students quickly makes students aware of key concepts often overlooked (or taken as given) in traditional writing workshops. A partial list of these would include the 'materiality' of digital text, the layered/networked ontology of text, and the evolving function of authorship (each discussed below).

This is not to say that so-called *analog* writing workshops cannot address these concepts (for example, as in some experimental writing workshops), but only that in generative writing workshops it is quite difficult *not* to address such issues as they emerge directly from one's engagement with the technical systems that mediate the writing process. These issues often emerge as various technical media "push back" against the intentions of the writer, making apparent the particular *affordances* of a specific medium. Writers, especially

for those new to digitally-mediated writing, are compelled to focus not only on content, but also context, form, and the constructed nature of writing itself. As a result of such "push-back," one student became more conscious of her tendency to think of art as "pristine" and removed from complex "variables" that structure and construct it: "I'm a little distressed to find that I may consider art to be more pristine than I thought I did, more removed from a messy, complicated set of variables that contributed to its making."

Especially for those students self-identifying as "writers" (as opposed to digital artists or programmers), the push-back experienced with RiTa (and all technical systems to varying degrees), helped displace assumptions concerning the immateriality of text. Moreover, this "push-back" seemed beneficial for all writers, even those not planning to further pursue work in programmable media. The same student quoted above wrote:

The work we've done in this class has been surprisingly generative for me in terms of thinking about my writing practice in general; surprising because I didn't expect tools and practices of "electronic writing" to bleed over into "regular writing."

The expectations of this student reveal the common assumption of a divide between digital and traditional writing practices. Yet the unexpected surprise (a notion we return to below) is that "digital" writing can substantially – and productively – influence "traditional" writing; a theme that emerged repeatedly in student comments throughout the workshops.

In our experience, when students experience the "push back" of the medium (in this case an explicitly technically one), they directly confront the apparatus surrounding their writing practices and thus become conscious of the medium's materiality. While a different technical apparatus surrounds traditional writing practice – the use of word processors, the printing of text on paper, publishing and distribution systems, etc. – the embedding of language within various material supports is generally overlooked in favor of some conception of a text as an abstract entity not requiring material instantiation or easily ported between differing material supports. We need not delve deeply into this topic as N. Katherine Hayles (and others) have argued extensively and elegantly for an awareness of textual materiality. Indeed, Hayles has argued for a new methodology of interpretation called Medium Specific Analysis, "a kind of criticism that pays attention to the material apparatus producing the literary work as a physical artifact" ^[29]. The simple point we wish to make is that ideas central to such an analysis emerged quickly and naturally in the courses taught at Brown as students were immediately confronted with the problem of designing the interface for their texts. One writer described his realization that "a programmer needs to pay attention to every design choice (s)he is making in order to make sure that these choices are in alignment with the overall artistic idea." Indeed, the fact that students working with RiTa had to invoke methods to set the window size and background color, the text font, its color, size, and placement on the screen made them immediately aware of both the malleable materiality of code and the dynamic materiality of the screen on which the text appeared. In addition, this awareness of the materiality of the digital text influenced many writers' understanding of traditional print media. As one student wrote:

[This class] made me a more visual person; I wasn't as much a visual thinker as I am now. Therefore, the experience of electronic writing made me sensible to the use of space on a sheet [of paper]. I will continue writing traditionally while being sensitive to the use of fonts, sizes, images and colors, while paying attention to the layout of a text.... Writing electronically opened my eyes to those details.

LAYERING

The RiTa library provided students with a range of literary functions designed to complement the visual focus of the Processing environment. A range of assignments with RiTa tools further engaged students in "writing" on a number of different levels, from initial sketches and textual drafts, to formalizations of literary parameters within a grammar or combinatoric scheme, to more traditional programming in a formal language, Java. Exposure to this multiplicity of layers continually reminded students of the layered nature of textual activity that surrounded their practice – from the code of various libraries and programming languages, to the specificities of the operating system and network protocols, to the design of the web-browsers that provided the concrete frame for their work. It was readily apparent that design decisions at these (often "invisible") layers had significant ramifications for the concrete writing practices of the students.

Within this sedimentation of technical layers, a key question that repeatedly arose concerned the location of the text. Was it in program code of the students? Or the text files that specified the grammar and lexicon to be used? In the RiTa and Processing code that combined and executed the above? In the *output* text that appeared onscreen at runtime? It was evident from student presentations and critiques that multiple layers of "text" existed. Interestingly, one feature of both Processing and the RiTa library was that source code and grammar files ^[30] were published (by default) along with the finished piece. As such, we were able to easily read at multiple levels below the "surface" in our critiques. Most students thus felt the text to be some amalgam of the three layers they had written (surface, grammar, code) rather than simply the output to the screen that varied from run to run of the programs. When questioned more closely most students felt that *all* the software tools employed – from RiTa to Processing to Java, to the network protocols and operating system code – could be conceived as potential texts for analysis, though each contributed to the final output in varying degrees. One student said the following concerning the grammar layer of the text:

I guess theoretically the grammar exists independently as a piece of writing that, like most language on a page, does nothing, but I know when I look at a grammar file that it has a programmatic counterpart. It goes somewhere; it does something; it will change based on a set of rules it defines. Which is all to say, I don't know how I feel when I look at a grammar. I certainly consider it to be part of the text of a piece, but I also can't separate it from its use value, which is not to say, oh, it's an inferior piece of writing, just that I'm not sure how to categorize it or interpret it *as itself*.

Such questions, concerning how to "interpret" the code and grammar layers, led to productive discussions concerning the nature of sedimented texts in general. The student's uncertainty in "how to categorize" the grammar "as itself" was an important step toward understanding that individual textual layers do not possess an "as itself," but always exist in relation to other layers. Yet at the same time, this tangible separation allowed us to analyze the different layers *as if* they were stand-alone entities. For example, due to students' natural tendencies to make associations with surrounding text when writing, there were often further level of poetic association present in the grammar files that were *absent* from the program's output. In this light, close-readings of the grammars themselves proved quite interesting, as if the set of possible lexical choices for each grammatical rule constituted small poems in themselves. Below follows an excerpt from one student's grammar file, the "|" symbols signify OR, so that for any run of the piece, *only one* of the lines below would appear, yet relationships *between* the lines are often clear, and at times, quite interesting.

and how he came to know the truth |
is a dubious gesture, and one not to be trusted |
as the boys' voices reached down through the floorboards |
douglas fir we think, though we can't be sure |
in concentric circles | with a passion typified by adolescent lust |
and if it rang, did we pick it up? | they knew each other almost certainly |
if the door had opened, it would have showed something completely...

One could read the first three "phrases" of this excerpt as a complete sentence, with the first two lines expressing a coherent thought. It is also difficult not to forge connections between elements such as "the floorboards" and the "douglas fir," or between the "boys' voices" and the clause "with a passion typified by adolescent lust." One also observes in this selection that the text seems to grow increasingly fragmented toward the end, as if the writer was breaking free from composing in a linear fashion and allowing the phrases to unhinge (the image of the ringing telephone, for example, appears unconnected to any other parts of the excerpt).

THE AESTHETICS OF GENUINE SURPRISE

A central aspect to the generative approach is the use of an externalized system, created by the artist but rarely completely under her control. Standard software tools are deterministic systems that always produce the same results, while generative systems are dynamic processes that must be harnessed and even farmed. The artist specifies the initial boundaries and strategies of creation, and then enters into a feedback loop of adjusting parameters in a search for optimal regions in parameter space. The moment of genuine surprise is often the moment of breakthrough ^[31].

Much generative art seeks out such "moments of genuine surprise" as a perceived condition of an artwork's success. Indeed, the experience of surprise is a key category of generative art, and one might aesthetically judge a work of generative art in terms of whether it generates surprising visual forms, sounds or language (though much visual generative work seems to use traditional criteria of aesthetic "beauty," perhaps derived from the traditions of abstract art). Surprise concerns the unexpected, a movement away from the

convention of old forms and the ordinariness of everyday language. In terms of reading and experiencing generated works in the classes at Brown, the discourse of the unexpected often emerged during critiques and also appeared in student responses to their activities. If a work generated unexpected results it was often perceived to be more engaging and successful. One student wrote that "in particular it has been kind of revolutionary to think about introducing other source texts into my writing, to scramble them, to come back to them during the writing process, to be surprised at what meaning emerges." This feeling of surprise seemed a key component of students' experience as they explored generative writing i increasing depth.

Interestingly, Marius Watz employs the phrase "genuine surprise" to characterize the success of generative works. Here, the genuine is a marker of the *authentic*. The poet Charles Hartman makes a similar connection between surprise and authenticity: "Part of my hope is to surprise the reader; part of it is to surprise myself. The idea isn't just to make the process of [generative] writing more entertaining but to *authenticate* it. If I'm discovering, the reader is more likely to have a sense of discovery" (italics in original) ^[32]. For Hartman, the production of surprise operates as a kind of proof to the reader (whether himself or another) that the generated text is "worth something." Surprise gives the artwork credibility in the eyes of the reader, perhaps usurping the credibility that the "author function" has tended to bestow on the literary text.

The experience of surprise, or *being taken* by surprise, is strongly linked to an absence of control. Indeed, surprise (*sur-prehend*) literally "overtakes" us. Surprise is thus perceived as a temporary loss of subjectivity, as a relinquishment of one's subjective intention, either to another's control or to objective forces beyond one's control. Some students experienced this loss of control as liberating:

...the enjoyment and generativity is to be found in the unexpected matches between words and phrases, between things I would have never thought to put together, but that manage to make a kind of sense beyond themselves. I like the idea of a piece of writing creating its own meaning, because it means I don't have to try to do it. I can write in a field as opposed to writing with a perceived trajectory.

"Genuine surprise" marks the reader's confrontation with a complete lack of control. And again, the word "genuine" returns us to the tangle of words in the etymological family of generation, engine, and genius. According to the OED, originally the word "genius" referred to a "tutelary spirit" (a *genii*) that controlled one's fate and "determined his character." It was an "other mind" inside of our own, a guiding, mystical force beyond our control. The importance of the identification of surprise is that it moves us beyond the binary of "genius vs. engine" toward a conception of artworks as fields for potential surprise regardless of their human or machinic origins. Moreover, in terms of teaching generative art practices, identifying surprise as a key aesthetic category helps to critically foreground important questions; e.g., What is the nature of surprise? What causes surprise in the viewer of a generative work? Is surprise caused by our disbelief that programmed systems can produce emotional effects on par with those created by humans? Further, if we can discuss surprise in terms of the human response of an artwork, might we also discuss what surprise might result from a "machinic reading" of a work? Finally, we must keep in mind that "surprise" is also a historical and discursive *construction*. Surprise is not a mystical category beyond explanation (like the

"tutelary spirit" that supposedly guides us in acts of genius), nor is it an appearance of the radically and completely new, the totally unexpected. Rather, surprise and the unexpected are historically-constructed categories. It is well known that capitalism and the ongoing process of modernization continually cultivates the unexpected and the surprising (e.g., the focus on new product innovation). Indeed, modernism and the avant-garde are often driven by Pound's exhortation to "make it new!" Thinking critically about the aesthetics of surprise requires an understanding of its construction and mobilization in contemporary culture. Getting students to think about surprise as a central aesthetic criteria for generative art is a good first step, but persuading them to examine the origins of such surprise is an even more productive move toward inspiring work that *critically* engages such an aesthetic.

ROUGHING IT UP

"What I know is that I felt free when I was writing my grammar, freer than I've felt in writing anything for a while, because I knew it wasn't going to appear in the order I wrote it. I knew my thought-order would be disrupted, so I could say anything I wanted – talk about my grandpa, my feelings, how I stubbed my toe – and those boring and self-indulgent journal-writing tendencies would get roughed up by some other type of language. Language I'd also written, of course..."

We tend to think of rough drafts as needing to be refined, molded, shaped, smoothed, and polished. Conventionally, the rewriting process is an un-roughing of content and form. In teaching generative literature we have found it a useful thought-experiment to turn this thinking upside-down. One can think of the algorithmic process that operates on a source text as creating a "draft" that blows through the original text and "roughs it up" once again. The rewriting that the algorithmic process produces would thus become a re-roughing of a text, a distortion or a deviance from some original or from ordinary language use ^[33].

Focusing on the production of "rough drafts" as works in themselves draws attention to process rather than finished product. Thus temporal, running processes can become the "finished" artwork. This is related to what Peter Lunenfeld has called the "aesthetics of unfinish." He writes,

...to celebrate the unfinished in this era of digital ubiquity is to laud process rather than goal – to open up a third thing that is not resolution, but rather a state of suspension. To get to that unresolved third thing – that thing in abeyance – we need first to acknowledge the central effects the computer has had on art and culture ^[34].

It is the possibility of "finishing" that is forever put aside as the process enters a state of suspended animation. In many student projects we witnessed such an "aesthetics of unfinish:" texts where the reader could endlessly click the mouse in order to generate new material to read; a web TV application where one could choose a channel and watch endless images (gathered from YouTube) with voiced computer commentary (generated via statistical methods); a Twitter-based application that automatically and

endlessly generates updates via algorithmic processes and context-free grammars. Although generative art practices can be used as aids in the formation of static finished texts, the roughing-up of processes can also become the unfinished text itself.

As the student quoted above states, the roughing up of her text was accomplished through "some other type of language." Awareness of this "other language" can lead students to think about the similarities and differences between these types of language — specifically natural vs. programming languages. If the algorithmic process causes the emergence of "rough drafts" on the surface level of audience reception, one begins to ponder the possibility that the finished product is actually the combination of code, rules and processes that create these drafts. We would not want to argue that the "code *is* the text," but rather that by comparing the program code's static or "finished" nature with the malleability of the "unfinished" surface text we can make useful distinctions between varieties of encoded languages ^[35]. One student, attempting to generate short political speeches using State of the Union addresses as his source material, reflected on the "rough" outcomes of the process: "That the resulting text remains babble is a testament to the limitations of tagging language by part of speech." While there may be some truth in this assessment, it is perhaps more interesting to position such limitations as locations of rhetorical significance, ripe for aesthetic exploration. Within the domain of generative literature, one can attempt to generate grammatically correct language, but one can also seek out the places where this attempt fails or becomes difficult, where the technical limitations of encoded languages butt up against the conventional, grammatical "limitation" of natural language. Indeed, in light of the real-time constraints of the RiTa components — implemented not always with the most *correct* algorithm but rather, in order to satisfy the design constraint of web-based execution, the lightest and fastest — the locus of technical breakdown is often as interesting as any instance of an adequately-mimicked linguistic "realism."

SITUATING THE AUTHOR

But in every workshop I've participated in so far it's been drilled into me to present what I've written and say nothing as I listen to people's responses; no one wants to know what I "meant" to do or what I almost did or what I did and then deleted. I've been taught that my knowledge of what contributed to the making of a piece actually interferes with other people's ability to judge it. So a discourse in which process and product are discussed or approached simultaneously is still a little weird to me.

There is a sense in the above quotation from one of the students that analog writing workshops have recently tended to obscure the author function and the question of authorial intent. Partially, this can be attributed to the "death of the author" thesis and the rise of the *text* over the *work* in post-structuralist thought ^[36]. Yet, even though the position of the author recedes in traditional workshops, they ultimately remain the guarantor of the text's meaning (as when, at the end of a workshop critique the author is often given space to express his or her intent). In the study and practice of digital writing though, the question of the author's position, the nature of her intent, forcefully returns as a productive component in the practice of generative writing. As one student put it, "the issue of authorship rears its head every time we talk about

machines doing the work of humans." When machines become involved, the dance between genius and engine generates new discourse, and many (if not all) of the students in the courses taught at Brown were interested in the question of authorship. For example, one student stated that she sometimes forgot the author when interpreting digital texts and wondered what the program was attempting to express, yet, she also "kept in mind that there is an author's intention lying behind [the program]." In a stronger statement, another student wrote, "I will consider my software as an extension of my intention: if I had the time and patience, I could do it by hand. Software is my executor, it does what I would do if I was there." Interestingly, some of this student's work used the machine to profoundly interrogate the intersection of subjectivity and language, of authorship and the daily production of text ^[37]. The prominence of the problematic of the author can be seen in a range of student projects that blurred boundaries between human and machine subjectivity: a Mozilla plug-in that automated web surfing and attempted to simulate a particular machine's desires through the content of its generated searches; a robot that took images of humans and generated text according to an analysis of the image's properties; an automated "blog" generator that simulated the activities of a non-existing human through blog and Twitter postings.

In the context of this paper, what is important is that the machine generation of texts invokes a palpable anxiety concerning the relationship between meaning, authorship, and text. Building on this productive uncertainty, the practice of generative writing offers a unique opportunity to re-engage aesthetic questions of authorship and meaning. One student noted how useful it was to constantly locate "her boundaries" in relation to how much meaning she had placed into the text, how much control she could assert or give up. This negotiation of boundaries reveals a constant critical engagement with questions of authorship. The awareness that the problematic notion of authorship is still very much alive is beneficial, we argue, for examining important questions for contemporary practice, e.g., appropriation, creative plagiarism, and even machinic authorship. The question often arose concerning the best use of a writer's time when working with generative text; on the lyricism of one's own "inspired" words (the source text for a work) or on the processes that may connect and recombine them? Moreover, when one begins to impute "subjectivity" or authorship to the machine one begins to think about how the machine might express its own "voice" using natural language. If the machine is an author, does the machine have a style that may express its individuality? One student's work – often in collaboration with others – investigated the productive tangle of "machine subjectivity" throughout the semester. These intriguing projects included a short program that generated mundane, everyday "algorithms for human behavior" (supposedly created by a machine conscious of its human user); a program that searched the system logs on other students' laptop computers, occasionally using the results to formulate English phrases, that the computer would then audibly speak to the user, reflecting its internal 'emotional' state; and, as already mentioned, a web browser plug-in that allowed the machine to take control of its user's browser, performing actions that revealed its programmed desires and "personality."

READING THE ALGORITHM

Roland Barthes once wrote, "In the multiplicity of writing, everything is to be disentangled, nothing deciphered; the structure can be followed, 'run' (like the thread of a stocking) at every point and at every level, but there is nothing beneath: the space of writing is to be ranged over, not pierced" ^[38]. In Barthes' estimation of textuality, there is no depth, only the surface of structure, that the reader "runs" as they

produce meaning from the structure of the text. Yet in digitally-mediated writing, textual depth reappears as the "running" code or process, often perceived as operating beneath (and/or producing) the surface representation; although, we might say that the code "pierces" the text when it executes. The reader, for Barthes, "runs" the text or, as he says in his essay "The Death of the Author," the reader "set[s] it going," producing the event of the text. Yet, the digital text is already "run" by the algorithmic processes that set the text "going." The point to consider here is that in digital writing there is always already a prior "reading," a machine reading which necessarily occurs before the human reader experiences the text. Since it is now commonly argued that the author function has been, at least in part, re-situated as the figure of the programmer — especially in generative systems — it is she who actually produces this potential "reading" through the programming of a process (how the machine will "interpret" the data). In fact, one could argue that the question of authorship reappears here precisely because the author as programmer usurps the position of the reader, offering to the traditional reader an already programmed reading of a data set.

Extending this notion, the algorithm can thus provide a privileged location of reading, specifically an *authorial* reading; one that other readers, generally focusing on the output of the algorithm, can only strive to understand. In fact we might imagine a future in which readers of generative texts have become so adept at reading processes that such an observation might appear obvious. And indeed, the work of interpreting and understanding the rhetoric of processes has already begun; in the research of Noah Wardrip-Fruin, for example, and in the emerging field of Critical Code Studies^[39]. When dynamic processing is no longer unfamiliar but rather a common attribute of a text (either readable through surface effects or through provided source code), then the question of authorship will likely once again subside while the "power" of the reader may well rise. It is almost as if readers — in the process of learning how to read differently, how to read processes — need the author as guarantor of an existent meaning until they develop the critical apparatus necessary for reading works of literature in programmable media. Similar to how multiple readings of a text can displace the privileged meaning of an author's intent, in learning to read (and interpret) processes, the algorithm as the locus of meaning (an idealization at best) will likely become simply another location in which meaning can reside. Of course, the work of learning to read algorithmic processes is still in its infancy, as is the understanding of how processes can have rhetorical effects. Helping students to see the "formal rules," algorithms and dynamic processes they author as a type of "reading" may well help them to think more closely about the rhetorical significance of what they create. Just as students in literature classes are taught to "read" texts and formulate arguments through this reading, one can think of algorithms as "arguments" or points that one is trying to express. In either case it is certain that the various practices of "reading" within and through algorithms will be an essential skill for the next generation of writers and critics, whatever the level of their engagement with technology.

CONCLUSION

In the course of this paper we have presented and discussed a range of issues — our "lessons learned" — stemming from student reflections as they engaged with the RiTa toolkit and the practice of generative literature. We have attempted to accomplish three goals throughout: first, to identify a set of potentially productive concepts for students, teachers and practitioners of generative art; second, to forge a connection between the emergence of these concepts and the development of technical tools (or engines) like RiTa; and third, to critically examine these concepts in hopes of both challenging and extending current thinking on

the aesthetics of generative art. All of these concepts — from generative pedagogy to new complications in authorship, from the aesthetics of surprise to the ripe categories of layering and process — will certainly be "roughed-up" as more practitioners and theorists turn their gaze toward generative aesthetics. While predicting the dynamics of literary practice remains beyond the grasp of even our most discerning algorithms, the experience of developing RiTa and using it within a classroom environment suggests that engagement with programmable media will continue to profoundly influence contemporary aesthetics. It is our hope that the discussion above will help extend this influence into the domain of the literary.

REFERENCES AND NOTES

1. See «<http://rednoise.org/rita/documentation/index.htm>».
2. 'Electronic Writing' (LR0021) and 'Digital Arts & Literature' (CSCI1950) at Brown University. The various iterations of these courses were open to both undergrads and graduates and focused on the analysis and creation of literature in programmable media, with the former sponsored by the Literary Arts Department and the latter sponsored jointly by the Computer Science Department and the Rhode Island School of Design's Digital+Media program. Students backgrounds were highly varied with a diverse mix of creative writers, plastic and digital artists, computer scientists, and digital media theorists. Technical backgrounds varied from programmers with many years of experience to those with zero.
3. Daniel C. Howe, "RiTa: a Generative Language Toolkit for Digital Literature," Unpublished Manuscript. 2008. «<http://mrl.nyu.edu/~dhowe/words.html>»
4. As Philip Galanter has attempted for the category of 'generative art'. Philip Galanter, "Generative Art and Rules-Based Art," *Vague Terrain* 03 (June 2006).
«http://philipgalanter.com/downloads/vague_terrain_2006.pdf»
5. Christiane Fellbaum, ed., *WordNet. An Electronic Lexical Database* (Cambridge, MA: MIT Press, 1998).
6. Casey Reas and Ben Fry, *Processing: A Programming Handbook for Visual Designers and Artists*, (Cambridge, MA: MIT Press, 2007). See also «<http://processing.org/>»
7. B. Shneiderman, "Creativity Support Tools," *Commun. ACM* 45, 10 (Oct. 2002), p. 116-120. DOI= «<http://doi.acm.org/10.1145/570907.570945>»
8. Jean-Pierre Balpé, "Principles and Processes of Generative Literature: Questions to Literature," *Dictung-Digital* (January 2005). «<http://www.brown.edu/Research/dichtung-digital/2005/1/Balpe/index.htm>»
9. Philip Galanter, 2003 "What is Generative Art? Complexity Theory as a Context for Art Theory".
«http://philipgalanter.com/downloads/ga2003_paper.pdf»
10. Marius Watz, "Fragments on Generative Art," *Vague Terrain* 03 (June 2006).
«<http://www.vagueterrain.net/content/archives/journal03/watz01.html>»
11. See Philip Galanter, "Generative Art and Rules-Based Art," *Vague Terrain* 03 (June 2006).
«http://philipgalanter.com/downloads/vague_terrain_2006.pdf», and also Florian Cramer, *Words Made Flesh: Code, Culture, Imagination* (Rotterdam: Piet Zwart Institute, 2005).
«<http://pzwart.wdka.hro.nl/mdr/research/fcramer/wordsmadeflesh/>»
12. L. F. Menabrea, "Sketch of the Analytical Engine Invented by Charles Babbage," in Jeremy M. Norman, ed., *From Gutenberg to the Internet: A Sourcebook on the History of Information Technology* (historyofscience.com, 2005), p. 244.

13. Jim Carpenter, "Electronic Text Composition Project," The Slought Foundation. (2004). This short text accompanied the Slought Foundation exhibition entitled "Public Override Void" – an exhibition featuring Jim Carpenter's Electronic Text Composition Project. See also
[«http://www.slought.org/content/11207/»](http://www.slought.org/content/11207/)
14. For a discussion of the concept of distributed authorship, though neglecting the notion of invisible or unacknowledged authorship that we use here, see Christiane Heibach, "The Distributed Author: Creativity in the Age of Computer Networks," *Dichtung - digital* (August 2000).
[«http://www.brown.edu/Research/dichtung-digital/2000/Heibach/23-Aug/index.htm»](http://www.brown.edu/Research/dichtung-digital/2000/Heibach/23-Aug/index.htm)
15. From the WIDE website: "When we use the term 'digital writing,' we refer to a changed writing environment – that is, to writing produced on the computer and distributed via the Internet and World Wide Web. We are not talking about the computer as a stand-alone machine for writing; although that particular technological development has indeed changed the writing process, the computer itself as a stand-alone machine is not revolutionary in the sense we mean. Rather, the dramatic change is the networked computer connected to the Internet and the World Wide Web." Of course, advocates such as the WIDE collective are not addressing "digital writing" as a literary practice, but in terms of writing instruction and developing general composition skill sets for college courses (such as essay writing). Nevertheless, the danger remains that definitions of "digital writing," solely expressed in terms of distribution and increased connectivity, might eclipse a more radical understanding of digitally mediated writing where one must understand and engage the programmable machine. See
[«http://www.technorhetoric.net/10.1/coverweb/wide/kairos2.html»](http://www.technorhetoric.net/10.1/coverweb/wide/kairos2.html).
16. In this respect we agree with Jeannette M. Wing's call for the teaching of what she calls "computational thinking" – an "attitude" or way of thought that employs concepts and practices from computer science in order to teach critical and analytical skills to students. Jeannette M. Wing, "Computational thinking." *In Communications of the ACM* 49(3): 33-35 (2006).
17. E. Felton, "Free-dom to Tinker," (2008). See [«http://www.freedom-to-tinker.com»](http://www.freedom-to-tinker.com)
18. See H. Willis, "Toward an Algorithmic Pedagogy," *Fibreculture* 10 (2007).
[«http://journal.fibreculture.org/issue10/issue10_willis.html»](http://journal.fibreculture.org/issue10/issue10_willis.html). Also W. S. Seaman. "A Generative Emergent Approach to Graduate Education", in *Educating Artists for the Future: Learning at the Intersections of Art, Science, Technology, and Culture*. Intellect Ltd: 2008.
19. See [«http://www.generative.net/read/home/»](http://www.generative.net/read/home/). For information on the Generator.x exhibitions see [«http://www.generatorx.no/»](http://www.generatorx.no/). For information on the Autopilot exhibition see [«http://www.artificial.dk/articles/generativespecial.htm»](http://www.artificial.dk/articles/generativespecial.htm).
20. Florian Cramer, *Words Made Flesh: Code, Culture, Imagination* (Rotterdam: Piet Zwart Institute, 2005).
[«http://pzwart.wdka.hro.nl/mdr/research/fcramer/wordsmadeflesh/»](http://pzwart.wdka.hro.nl/mdr/research/fcramer/wordsmadeflesh/)
21. Interestingly, this student expressed a similar view to that of the film semiotician Christian Metz who, in his 1974 book *Film Language: A Semiotics of the Cinema*, offered the possibility of permutational poetry created by computers as a negative vision of the future – a dystopia that the cinematic image (based as on the photographic image) would resist. For Metz the cinema was not a language in the same sense that speech and writing were: the cinematic image was "a message without a code." The cinematic image had, in semiotic terms, neither a first nor second articulation. While spoken language is articulated into morphemes (a first articulation, e.g, words like "dog" or "window", or semantic modifiers like "un-", and "re-") and phonemes (a second articulation with basic units that do not carry significance in themselves), Metz argued that cinematic images do not have any basic units similar to

distinctive features or phonemes, nor does the simplest cinematic "shot" create a morpheme (e.g. a shot of a dog does not simply mean the word "dog" but something like the sentence "here is a dog," an assemblage of multiple morphemes). The photographic and cinematic are not decomposable into smaller units of manipulation and thus, thankfully in Metz's opinion, cannot be generated by a computer. Christian Metz, *Film Language: A Semiotics of the Cinema* (Chicago, IL: University of Chicago Press, 1991), p 31-91.

22. As an example, consider a photograph of a dog. Assuming one could effectively isolate different parts of the image (ear, tail, paw, leg, snout, eye, etc.) using such "primitives" as the raw material for generation would be likely only to create new image with a "fractal-like" or collaged appearance. For a contemporary example, see Cornelia Sollfrank's *Net Art Generator* – «<http://nag.iap.de/?lang=en> and <http://net.art-generator.com/>» – a generative work that creates collage-like products similar to what the student seems to have had in mind.
23. Marius Watz, "Fragments on Generative Art," *Vague Terrain* 03 (June 2006).
«<http://www.vagueterrain.net/content/archives/journal03/watz01.html>»
24. We should point out that digital images can also emulate photographic realism (think of the real time rendering of complex three-dimensional worlds in many video games) where pixels are arranged according to a mathematical 'grammar' based on Renaissance perspective and other interpretations of the mechanics of human vision. But this supports our argument since, as Watz points out above, one encounters relative few generative works employing figurative visual representations.
25. See «<http://www.nexttext.net/>».
26. See «<http://joerg.piringer.net/index.php?href=namshub/namshub.xml>».
27. John Cayley has written eloquently concerning how the "literal" was "always already" the digital in the sense that language is constructed through discrete units that can also be manipulated. He attributes the contemporary waning of literal manipulation and the waxing of manipulations concerned with image and sound to the ideological import of the "digital" that effaces the fundamental discreteness of language with a new discourse of digital discreteness (pixels, binary representations, etc.). Moreover, Cayley points out that the generation of language based on non-conventional grammatical forms, or forms that play with or displace conventional grammar "immediately evoke notions of legibility, error, and appropriateness; and any aesthetic effects of this literal programming may be stunned by these considerations." The ideological force of "appropriateness" and "error" that tends to oppose artworks that dismantle conventional grammatical form is indeed one reason why we are seeing less generation of text. John Cayley, "Literal Art," *Electronic Book Review* (November 2004).
«<http://www.electronicbookreview.com/thread/firstperson/programmatology>»
28. Such use could be in the form of word processing applications and their default-enabled warnings about spelling and grammar "errors," or through more sophisticated plot-generating programs, or the use of the Internet for various research purposes, dictionaries, thesauruses, etc..
29. Katherine Hayles, *Writing Machines* (Cambridge, MA: MIT Press, 2002) p. 29.
30. A RiTa "grammar file" is an external plain-text file that specifies the production rules and lexical entries for text generation via a (probabilistic context-free) grammar.
31. Marius Watz, "Fragments on Generative Art," *Vague Terrain* 03 (June 2006).
32. Charles Hartman, *Virtual Muse: Experiments in Computer Poetry* (Hanover, NH: Wesleyan University Press, 1996) p. 35.

33. There is nothing entirely new here, especially for those familiar with the traditions of surrealist writing practices, the cut-up techniques of Brion Gysin and William Burroughs, or the procedural methods of Jackson Mac Low, Emmett Williams, Charles Hartman and others. Of course many of these "roughed-up texts" were often revised in order to polish a finished text. In the genre of codework however – as practiced by MEZ, Talan Memmott, Alan Sondheim and others – it is the structures and syntax of programming languages that rise from the occluded depths to "rough-up" traditional, communicative language, but here as well the final work is generally a static product.
34. Peter Lunenfeld, *The Digital Dialectic: New Essays on New Media* (Cambridge, MA: MIT Press, 1999) p. 8.
35. As tools like RiTa can also be used to dynamically generate grammars (as several students in the course attempted) and program code itself (even self-modifying program code), we should note that techniques such as these could serve to "un-finish" the encoded text as well.
36. For the difference between the post-structuralist concepts of 'work' and 'text' see Roland Barthes, "From Text to Work," *Textual Strategies: Perspectives in Post-Structuralist Criticism*, ed. Josue V. Harari. (Ithaca, NY: Cornell University Press, 1979) p. 73-81. The "death of the author" thesis can also be traced back to the influential concept of the "intentional fallacy," championed by the literary school of New Criticism, in which the text (minus authorial intent) becomes a unified field for interpretation of over-determined meanings. Yet, such theories still privileged the autonomous, bounded work over notions of a text as a networked and historically contextualized entity.
37. See Cristobal Mendoza's *Every Word I Saved* (2006) at «<http://www.matadata.com/projects.php?id=1>»
38. See Roland Barthes, "The Death of the Author" (1977)
«<http://social.chass.ncsu.edu/wyrick/debclass/whatis.htm>»
39. See Noah Wardrip-Fruin's Dissertation entitled *Expressive Processing* at
«<http://www.noahwf.com/dissertation/>». For an introduction to Critical Code Studies see Mark Marino, "Critical Code Studies" *Electronic Book Review* (December 2006),
«<http://www.electronicbookreview.com/thread/electropoetics/codology>»

RELATED URLS

Daniel C. Howe: «<http://mrl.nyu.edu/~dhowe/>»

A. Braxton Soderman: «<http://thefollowingphrases.com/>»

RiTa Project Home: «<http://www.rednoise.org/rita/>»

RiTa project Gallery: «http://www.rednoise.org/rita/rita_gallery.htm»

CSCI1950 Course Home: «<http://www.rednoise.org/pdal/>»

LR00021 Course Home: «<http://www.rednoise.org/e-writing>»

DOI PERMALINK

<https://doi.org/10.20415/hyp/006.e04>

ABOUT

Daniel C. Howe is a digital artist and researcher at NYU's Media Research Lab where he focuses on the design, implementation and analysis of generative literary systems. In addition to a background in sound and visual arts, Daniel has master's degrees in Literary Art (from Brown) and Computer Science (from U. Washington). He is currently a Visiting Professor in the Computer Science Department at Brown University. His work can be found at mrl.nyu.edu/~dhowe/.

A. Braxton Soderman is a doctoral Candidate in the Modern Culture and Media Department at Brown University. Before coming to Brown he completed a MFA in Writing at the California Institute of the Arts where he participated in the Integrated Media Program. He is currently studying digital aesthetics, electronic writing practices, and videogames. His dissertation, in progress, links videogames to larger historical and cultural forces operating within modernity. His work can be found at thefollowingphrases.com.