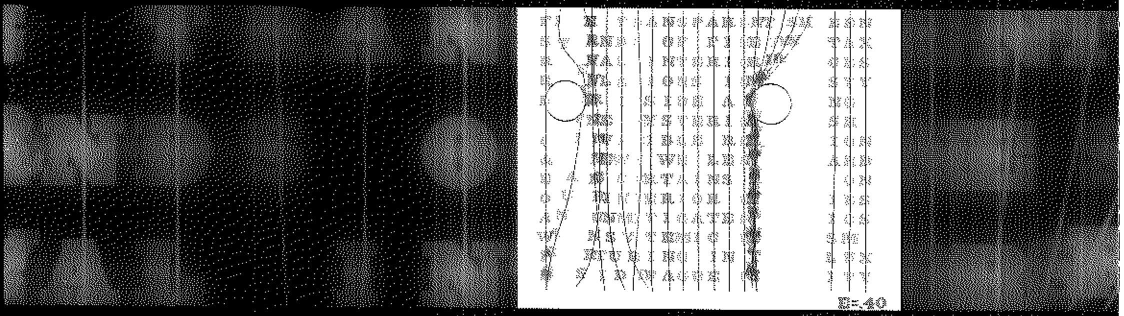
Roberto Simanowski
Jörgen Schäfer
Peter Gendolla (eds.)

Reading Moving Letters



Digital Literature in Research and Teaching

A Handbook

[transcript]
Media Upheavals 40

Noah Wardrip-Fruin

Five Elements of Digital Literature

1 Introduction

The terms "digital literature" and "digital art" are used frequently in our field, but rarely defined. When I use them, I mean something in particular by them. Let me begin by explaining how I use them.

A phrase like "digital literature" could refer to finger-oriented literature (fingers are "digits") or numerically-displayed literature (numbers are "digits") —but I mean "digital" in relation to computers, specifically as it appears in computer engineering phrases such as "stored program electronic digital computer." I mean literary work that requires the digital computation performed by laptops, desktops, servers, cellphones, game consoles, interactive environment controllers, or any of the other computers that surround us. I think that's what most of us mean, even if we've come to it in an ad-hoc way.

To take the other term in my initial phrase, "digital literature" could be used in the sense of "the literature" (the body of scholarly work on a topic) or it could mean particularly high-status writing—but I mean "literature" (and "literary") as a way of referring to those arts we sometimes call fiction, poetry, and drama (as well as their close cousins). I mean the arts that call our attention to language, present us with characters, unfold stories, and make us reflect on the structures and common practices of such activities. I should probably also say that I don't view the literary arts as a citadel, separate (and perhaps in need of defense) from, say, visual or performing arts. Much of the best drama, for example, brings together the literary, performing, and visual arts.

To me, "digital art" is the larger category of which "digital literature" is a part. It encompasses all the arts that require digital computation, not just the literary arts.

I write all this because, as an artist and scholar in the field of digital literature, I've begun to try to think more generally about the field of which I'm a part. I do this on one level because I'm curious about certain topics. For example: Where did this field begin? But at a deeper level I'm interested in questions of how we conceptualize what we make and study. The question that motivates my writing here is one of these: What might be a useful framework for thinking about the elements and context of works of digital literature? To put it another way: What do we need to read, to interpret, when we read digital literature?

2 Turing, Strachey, Love Letters

2.1 Turing Machines get Electronic

When I say that I mean "digital" as in "stored program electronic digital computer," what does that mean, more precisely?

In 1937 everyone who used the term "computer" knew what it meant. A computer was a person who calculated answers to mathematical problems.[1] These computers weren't expected to develop new, creative methods to prove outstanding mathematical problems. Rather, they were expected to follow a known and specified set of instructions which, together, formed an effective procedure for solving a particular kind of problem. We call such sets of instructions algorithms (from the name of Arabian mathematician al-Khwarizmi, Hillis 78).

But with the publication, in 1937, of Alan Turing's "On Computable Numbers" the world was quietly introduced to the mathematical thought experiment that we call a "Turing machine"—a concept that lay the groundwork for the kinds of non-human computers we have today. Turing's paper wasn't remarkable for imagining a machine that could carry out the work of human computers. In the 1930s there were already in operation a number of such machines (including Vannevar Bush's *Differential Analyzer*) and at least 100 years earlier (by 1837) Charles Babbage had conceived of an Analytical Engine, capable of mechanizing any mathematical operation.[2] Two things, however, separated Turing machines from all calculating machines in operation in the 1930s (and most of the 1940s) as well as all previous thought experiments (including Babbage's).

First, according to Turing's most prominent biographer, the Turing machine was developed in response to a mathematical question (posed by Hilbert) as to whether mathematics was decidable (cf. Hodges). That is, was there a method that could be applied to any assertion that would correctly determine whether that assertion was true? The Turing machine was a formalization that made it possible to discuss what could and couldn't be calculated—answering Hilbert's question in the negative, and establishing one of the primary foundations for computer science as a science (the investigation of what can and can't be computed).

Second, the imagined design of the Turing machine was in terms of a potentially implementable (if inefficient) mechanism. This mechanism was such that it could not only logically branch while following its instructions (doing one thing or another based on results to that point), and not only act as a universal machine (simulating the activities of any other calculating machine), but also store its instructions in the same read/write memory as the data on which it acted. This would make it possible, for example, for the machine to alter its own instructions while in operation. And it is from this type of capability that we get the words "stored program" in the phrase "stored program electronic digital computer." This lies at the heart of the computers we use each day.

This leaves us with the words "electronic" and "digital." The first of these can probably pass without definition—but the second is in need of clarification, especially given the mystifying ways in which it is sometimes used. Being digital, as it were, is not specific to computers, despite the fact that it's the word we've latched onto in order to represent computers. "Digital" information, as opposed to "analog" information, is represented by discrete rather than continuous values. It's actually related, according to the *Oxford English Dictionary*, to the sense of fingers and numbers as "digits." Each of the first nine Arabic numbers (or ten, if one includes zero) can be expressed with one figure, a digit, and these were originally counted on the fingers, or digits. Charles Babbage's Analytical Engine called for representing decimal numbers using ten-spoke wheels—which made it a design for a digital computer, because each of the ten wheel positions was discrete.[3] During World War II Konrad Zuse built the first program controlled digital computer that, instead of Babbage's decimal arithmetic, used binary arithmetic implemented in on/off electronics. This was a considerable simplification and made possible advances in increased speed and precision—important to our "digital" computers. Working independently (and very secretly) the British government cryptanalysis group of which Turing was part (and where he was instrumental in cracking the German Enigma code) created the *Colossus*, which has been characterized as the first fully functioning electronic digital computer.[4]

It was only after the war that a number of successful efforts were made toward stored program electronic digital computers. The first was the Manchester University *Baby* in 1948 (it used a CRT display for its storage) which was followed by a more complex Manchester prototype in 1949 and then replaced by an industrially manufactured version, the *Ferranti Mark I*, in 1951. Turing wrote the programming manual for the *Mark I* and constructed a random number generator that produced truly random digits from noise.[5]

2.2 Strachey's Next Step

Once there were stored program digital computers, all that remained (for our field to take its first step) was for someone to make literary use of one. I believe that—in 1952, working on the Manchester *Mark I*—Christopher Strachey was the first to do so.

Strachey went up to King's College, Cambridge, in 1935. While this is the same time and place where Turing was doing his fundamental work on computable numbers (as a recently-graduated junior research fellow) it is likely that the two knew each other only socially, and never discussed computing. Strachey worked as a physicist and schoolmaster after graduating from Cambridge, becoming increasingly interested in computing during the late 1940s. In January 1951 he was first exposed to a stored-program computer: the Pilot ACE computer under construction at the National Physical Laboratory. He began writing a program to make it play draughts (checkers), inspired by a June 1950 article in *Penguin Science News.*

That spring Strachey learned of the *Mark I* computer that had just been installed at Manchester—he had known Turing just well enough at Cambridge to ask for, and receive, a copy of the programmer's manual. He visited for the first time in July, and discussed his ideas for a draughts-playing program with Turing, who was much impressed and suggested that the problem of making the machine simulate itself using interpretive trace routines would also be interesting. Strachey, taken with Turing's suggestion, went away and wrote such a program, establishing his reputation immediately.

A year later, in June 1952, Strachey had wound up his responsibilities as a schoolmaster and officially began full-time computing work as an employee of the National Research and Development Corporation. That summer he developed—with some aesthetic advice from his sister Barbara, using Turing's random number generator, and perhaps in collaboration with Turing—a *Mark I* program that created combinatory love letters. This was the first piece of digital literature, and of digital art, predating by a decade the earliest examples of digital computer art from recent surveys (e.g., Paul).

Strachey described the operations of this program in a 1954 essay in the art journal *Enounter* (immediately following texts by William Faulkner and P. G. Wodehouse):

> Apart from the beginning and the ending of the letters, there are only two basic types of sentence. The first is "*My*—(adj.)—(noun)—(adv.)—(verb) *your*—(adj.)—(noun)." There are lists of appropriate adjectives, nouns, adverbs, and verbs from which the blanks are filled in at random. There is also a further random choice as to whether or not the adjectives and adverb are included at all. The second type is simply "*You are my*—(adj.)—(noun)," and in this case the adjective is always present. There is a random choice of which type of sentence is to be used, but if there are two consecutive sentences of the second type, the first ends with a colon (unfortunately the teleprinter of the computer had no comma) and the initial "*You are*" of the second is omitted. The letter starts with two words chosen from the special lists; there are then five sentences of one of the two basic types, and the letter ends "*Yours*—(adv.) M. U. C." (26-27)

As Jeremy Douglass notes, the love letter generator has often been discussed in terms of queer identity, rather than in literary terms. Certainly there are reasons for this—Turing and Strachey were both gay, and at least Turing openly so.[6] It might also seem from the most widely-reproduced outputs of the generator (e.g., that found in Hodges, 477-478) that it was a love-letter generator that "could not speak its name" (the word "love" being conspicuously absent).[7] But I suspect that the primary reason for the lack of literary discussion of Strachey's generator is that the output simply isn't very compelling. Here, for example, are the two outputs reproduced in *Encounter*:

> Darling Sweetheart
> You are my avid fellow feeling. My affection curiously clings to your passionate wish. My liking yearns for your heart. You are my wistful sympathy: my tender liking.
> Yours beautifully
> M. U. C.
>
> Honey Dear
> My sympathetic affection beautifully attracts your affectionate enthusiasm. You are my loving adoration: my breathless adoration. My fellow feeling breathlessly hopes for your dear eagerness. My lovesick adoration cherishes your avid ardour.
> Yours wistfully
> M. U. C. (26)

I would like to suggest, however, that examination of individual outputs will not reveal what is interesting about Strachey's project. As he wrote in *Encounter*: "The chief point of interest, however, is not the obvious crudity of the scheme, nor even in the ways in which it might be improved, but in the remarkable simplicity of the plan when compared with the diversity of the letters it produces" (27). That is to say, Strachey had discovered, and created an example of, the basic principles of combinatory literature—which still lie at the heart of much digital literature today (and, less commonly, non-digital works). Combinatory techniques allow a relatively small number of initial materials to be arranged, following certain rules, into a vast number of possible configurations. In relatively unconstrained systems such as Strachey's, each individual output is more likely to induce a humorous reaction than deep literary consideration. In fact, Turing biographer Hodges writes of the love letter generator that "Those doing real men's jobs on the computer, concerned with optics or aerodynamics, thought this silly, but . . . it greatly amused Alan and Christopher" (478). In the amusing nature of individual outputs, Strachey's system could be said to anticipate Roger Price and Leonard Stern's non-digital *Mad Libs* (conceived in 1953, but not published until 1958, cf. Montfort, "Literary Games"), though the love letter generator's more restrained combinatory vocabulary made it possible for most (rather than only a few) words to change from output to output. It is clear, however, from Strachey's contribution to *Encounter,* that he also understood the other side of combinatory literature—the view of the system itself when one steps back from the individual outputs, the remarkable diversity that can be produced by a simple plan. The production from such a simple plan, as has been pointed out with other combinatory texts, of more potentially different outputs than any of us could run our eyes across in a lifetime devoted to reading its output (Aarseth, "Nonlinearity and Literary Theory" 67). It is a work that can only be understood, in fact, as a system—never by an exhaustive reading of its texts.[8]

And it is not surprising that Strachey's effort is mostly of interest in terms of how it operates, rather than in the text it produces. After all, designing interesting ways for computers to operate—algorithms, processes—is at the heart of what most computer scientists and creative programmers do, from Turing and Strachey's moment to this day. In many ways we are al-Khwarizmi's descendants.

3 Data versus Process

3.1 What's a Computer for?

ACM SIGGRAPH, the Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques, holds an unusual yearly meeting. Tens of thousands of people come to see a combined industry tradeshow, scientific conference, and art gallery. One thing that is particularly striking is the differing status of images—of the products of computer graphics—in different parts of the conference. In the conference portion of SIGGRAPH, dominated by paper presentations from computer scientists, images play the role of examples and illustrations. Images are there to help explain the real results being reported—which are novel techniques, algorithms, processes. In the art gallery, while there are a few talks, most of the presentations are of art works, and most of the works are prints hung on the wall. In these prints, the images are not aids to an explanation of results—they, themselves, are the results. This can lead to some tension, because the artists know

it would be impolite to call the images made by the scientists naïve and uninspiring, and the scientists know it would be impolite to call the processes used by the artists trivial and uninteresting. And such tensions aren't unknown in the field of digital literature; for example, around the literary readings held at ACM Hypertext conferences. But they can also take a somewhat different form.

Let's take a step back. Writers over the last century have often wanted to exceed the limitations of black and white text printed and bound into a traditional codex book. For example, twenty years before William Faulkner's text preceded Strachey's in *Encounter,* the 1934 *Publishers' Trade List Annual* carried a listing for a book it had also listed in 1933, but which never appeared: a four-color edition of Faulkner's *The Sound and the Fury* (cf. Meriwether). Faulkner wanted four-color printing in order to make the time shifts in the first section of the book easier for readers to follow. He worried that alternation between italic and roman type might not be sufficient, and rejected suggestions from the publisher that he felt would break the flow of the language (e.g., space breaks). But such four-color printing would have made the book, even after it was a proven critical success, prohibitively expensive for many purchasers—especially during the 1930s U.S. economic depression. With a modern computer, of course, it wouldn't be. The additional cost of transmitting colored text over the Web (an HTML file made slightly larger by tags indicating sections that should be different colors) is negligible, and the ability to display color is already present. Combining images with one's text consumes a bit more transmission bandwidth, and sounds and moving images a bit more—but for those connecting to the Internet from businesses, universities, or high-speed home connections, the difference is barely worth comment. Finally, there may be some software cost—such as for a program like Flash or After Effects, if one wants the text itself to animate—but this is generally much less than the cost of the computer itself, and miniscule compared to color printing on paper or film. This has opened vast possibilities for writers, and taken literary approaches such as animated text poetry and fiction from occasional curiosities to international movements with communities in South and North America, Asia, and Europe. The results also, apparently, have wide appeal. For example, the 2002 *Dakota,* by Korean duo Young-Hae Chang Heavy Industries, has found an audience ranging from visitors to the Whitney Museum to browsers of popular online animation forums such as Albino Blacksheep.[9]

But from a computer science standpoint most of this work is utterly trivial. The vector animation techniques on display in *Dakota,* for example, are so well-understood that they are packaged into a mass-market tool like Flash. Other uses of computers by writers are similarly uninteresting on an algorithmic level. For example, take the area of email novels. While writers and literary critics may see a vast difference between the playful medieval sci-fi of Rob Wittig's *Blue Company* and the traditionally-structured titillating mystery of Michael Betcherman and David Diamond's *The Daughters of Freya,* on a process level they are exactly the same—human-written chunks of text sent to readers via email at timed intervals.[10]

For their part, computer scientists can claim some literary successes. The interactive character *Eliza/Doctor,* for example, was created by Joseph Weizenbaum at MIT in the mid-1960s—and has been continually read and ported to new computing platforms for four decades. Yet writers often find the work of computer scientists in digital literature quite puzzling. For example, while writers tend to assume that literary work focuses on the creation of language, Natural Language Generation (NLG) is only one area of interest for computer scientists working in digital literature.[11] Another is continuing the work on interactive characters begun with *Eliza/Doctor*—for which NLG may not be part of the research project, or which may not be experienced linguistically at all. Take, for example, the famous Oz project at Carnegie Mellon University. Their early 1990s *Lyotard* was a textual piece, presenting a simulated house cat with an intriguing personality living in a simulated apartment, in which the user read descriptions (to understand the world and the cat) and wrote them (in order to take actions and perhaps befriend the cat). While not a traditional form of writing, the output of the system is certainly recognizable to writers. But the Oz project's *The Edge of Intention,* which from a computer science perspective was part of the same research project on believable characters, featured no text at all—instead presented entirely as real-time animation (Bates). The same is true in the area of story generation. Some systems, such as Selmer Bringsjord and David Ferrucci's *Brutus,* are constructed with generation of literary text as an important part of their operations. But others focus entirely on generating story structures, with text output nothing more than a report on the structure, as is apparent in this simple example from Raymond Lang's *Joseph*:

> once upon a time there lived a dog. one day it happened that farmer evicted cat. when this happened, dog felt pity for the cat. in response, dog sneaked food to the cat. farmer punished dog. (139)

From this we could say that our stereotypes have been confirmed. Writers innovate on the surface level, on the reading words level—while computer scientists innovate at the process level, the algorithm level, perhaps without words at all. But as soon as this stereotype is expressed directly it also becomes apparent that it must be taken apart. And we have much more to point toward, for this purpose, than the work of those few writers who are also computer scientists.

We could begin, in fact, with the most cited example of combinatory literature: Raymond Queneau's *One Hundred Thousand Billion Poems* (1961). This work consists of ten sonnets, each of 14 lines. While one might expect, then, that this work would be more suitably titled *Ten Poems,* there is something in the construction of each poem that causes the number of potential poems to be much larger than ten. To wit: a reader can construct alternate poems by reading the first line of any of the original sonnets, followed by the second line of any sonnet, followed by the third line of any sonnet—and find that the whole work is artfully constructed so that any reading of this sort produces a sonnet that functions syntactically, metrically, and in its rhyme scheme. And here we see combinatory literature as (independently) discovered by a writer. Strachey's generator contains many more possible variations in each few lines of output, but there need be nothing artful in the selection of words—a thesaurus search for terms related to love will do the trick. Queneau's *Poems,* on the other hand, is a high-wire act of writing. He has created a process, but a process that only works when real attention is given to the words.

And Queneau, as a writer inventing processes—whether carried out by the reader or the writer—was far from unique. In fact, he was a co-founder of the Oulipo (Workshop for Potential Literature) in 1962, a larger group of writers and mathematicians that, to this day, continue such investigations. And we should not forget that, even before the Oulipo, 20th century literary practice already had been shaped by the process-heavy experiments of William S. Burroughs, the Surrealists, the Dada movement, and others. Clearly we need some way of framing these issues more accurately than "writers vs. computer scientists."

3.2 Crawford's "Process Intensity"

Chris Crawford, a noted computer game designer and theorist, writes about the concept of "process intensity":

> Process intensity is the degree to which a program emphasizes processes instead of data. All programs use a mix of process and data. Process is reflected in algorithms, equations, and branches. Data is reflected in data tables, images, sounds, and text. A process-intensive program spends a lot of time crunching numbers; a data-intensive program spends a lot of time moving bytes around.

For our purposes, this distinction between process-intensive and data-intensive maps nicely onto the inappropriate stereotype distinction between writers and computer scientists. Crawford's terminology is a more accurate and useful way of talking about these differences in approach to digital literature. When a work of digital literature emphasizes words, images, and sounds, those are all data. When it emphasizes algorithms and calculations, those are process.

And here, I believe, we come to the first part of an answer to the question posed in this essay's introduction: "What do we need to read, to interpret, when we read digital literature?" We must read both process and data. This is true, of course, not only for work in digital literature, and not only for the writers cited above for their innovation at the process level, but also for composers such as John Cage, artists such as those associated with the Fluxus group, and dramatists such as Augusto Boal.[12] In all of these cases, we are interpreting works that emphasize data and process to differing extents (and employ them in differing ways) and which cannot be fully interpreted from a sample output.

But this isn't all we need to interpret.

4 Interactions

4.1 Turing Test vs. Turing Machine

While the term "Turing machine" is quite famous in computer science circles, in popular culture Alan Turing's name is more often associated with the so-called "Turing test."

Turing, however, actually proposed a game, rather than a test. In a 1950 article in the journal *Mind*, he proposed the "imitation game." This game has three participants: "a man (A), a woman (B), and an interrogator (C) who may be of either sex." During the course of the game the interrogator asks questions of A and B, trying to determine which of them is a woman. A and B, of course, do their best to convince C to see it their way—the woman by telling the truth, the man by "imitation" of a woman. The proposed game was to be played over a teletype, so that nothing physical (tone of voice, shape of handwriting) could enter into C's attempt to discern the gender of the other players based on their performances.

Turing then asked, "What will happen when the machine takes the part of A in this game?" That is, what will happen when a machine, a computer, tries to "pass" as female—rather than a man attempting to pass in this way—under the questioning of the human, C? Turing proposed this as a replacement for the question, "Can machines think?"[13] (Turing 434)

Turing's paper is important for a number of reasons. One, as Nick Montfort has pointed out (Wardrip-Fruin and Montfort 49), is simply that it proposed a linguistic, conversational mode of computing at a time when almost everyone thought of computers as number crunchers. For philosophers, the primary audience of *Mind*, it provided a specific, phenomenological formulation of the "problem" of machine intelligence. For the not-yet-born field of Artificial Intelligence it provided inspiration. But for our purposes it provides something much more basic: an early, clear instance of digital media conceived as an interactive experience.

Remember, Turing machines give us a way of thinking about what is computable—that is, what questions can we pose and receive an answer? But, as Peter Wegner and others have pointed out in recent years, much of the computing we do each day is not of this form. Rather than a posed question to which we receive (or fail to receive) an answer, interactive computing assumes an ongoing set of processes, accepting and responding to input from the outside world, and in some cases (e.g., airline reservation systems) with any ending considered a failure. Or, as Wegner puts it:

> Claim: Interaction-machine behavior is not reducible to Turing-machine behavior.
> Informal evidence of richer behavior: Turing machines cannot handle the passage of time or interactive events that occur during the process of computation.
> Formal evidence of irreducibility: Input streams of interaction machines are not expressible by finite tapes, since any finite representation can be dynamically extended by uncontrollable adversaries. (83)

That is to say, there is a real, definable difference between a program like the love letter generator and a program for playing the imitation game. The generator runs and produces an output, using its data and processes, and is completely representable by a Turing machine. But to play the imitation game requires data, processes, and an openness to ongoing input from outside the system that results in different behavior by the system—interaction—something for which at least some computer scientists believe a Turing machine is insufficient.

4.2 Forms and Roles of Computation

Of course, many of the most significant forms of digital literature involve interaction of some sort. Confusingly, in common discussion of digital literature some of these interactive forms have been defined using terms that specify system behavior, while others have been defined in terms of user experience. For example, "hypertext" is specified at the level of system behavior. A hypertext is a text that—according to the term's coiner, Ted Nelson—will "branch or perform on request" (by links or other means) (cf. Wardrip-Fruin, "What Hypertext Is"). On the other hand, "interactive drama" is a term for interactive digital literature that produces for users an experience related to theatrical drama—and how the system behaves while producing this experience is not specified. In fact, there is no reason that the experience of interactive drama could not be produced through a system that presents users with hypertext. However, we might have trouble discussing this work in digital literature circles, because we have become so accustomed to viewing "hypertext fiction" and "exploration-based fiction" as synonymous, even though the system behavior of hypertext does not specify that the user experience be exploration based.[14]

To read digital literature well, we need to be specific about system behavior and user experience—and explicitly aware that data's impact on experience is at least as great as process and interaction. Films and codex books, for example, mainly have very similar forms of system behavior and user interaction, but differing data produces a variety of user experiences. And while it seems true that the link-based hypertext interaction of systems such as Storyspace lends itself to exploration-based fiction, we also have some evidence that quite different "locative media" technologies (such as those used in Teri Rueb's *Itinerant*) are good platforms for exploration-based fiction, and link-based hypertext has shown itself effective for utterly different experiences of fiction (such as in Scott McCloud's "Carl Comics").[15]

In grappling with the various forms of digital literature, I believe we would also benefit from greater specificity about the forms and roles of computation involved in the works we are considering. One approach to beginning this effort would be to propose different distinctions and see what organizations of the field result—both those that run along and those that cut across the grain of our current intuitions.

For example, we could distinguish (1) between (a) digital literary works for which computation is required only in the authoring process and (b) those for which it is also required during the time of reception by the audience. In this case, (a) includes Strachey's love letter generator, computer-generated stories and books of poetry, and any literary prints hung on the wall at SIGGRAPH. We might call it "digitally-authored literature." Conversely, (b) includes *Dakota* if viewed on a computer screen, *Eliza/Doctor* and all other interactive works, email novels, and any literary uses of virtual reality Caves, web browsers, cell phones, game consoles, and so on. We might call this "digital media literature."

A different approach would distinguish (2) between (a) those works in which the processes are defined in a manner that varies the work's behavior (randomly or otherwise) and (b) those that contain nothing within their process definitions that leads to variation. In this case (a) again includes Strachey's generator, and story generators, but also all interactive works, while (b) includes *Dakota,* most email narratives, and so on. A rather different arrangement, which we might refer to as "computationally variable" and "computationally fixed" digital literature.

Within category 2a we could make a further distinction (3) between those that vary (a) without input from outside the work's material and (b) with input from outside. Here (a) includes Strachey's generator, and most poem and story generators, while (b) includes pieces that change based on the day's news, or user interaction, or other inputs. We might call these "batch-mode" and "interactive" variable digital literature.

And within 3b we could distinguish yet again (4) between those that vary with input (a) other than from humans, aware of the work, and (b) from humans aware of the work. Interestingly, (a) includes few works of digital literature, though it is an active area of digital art, including works that vary with network behavior (clients for RSG's *Carnivore*), the weather (John Klima's *Earth*), and the stock market (Lynn Hershman's *Synthia*); while it is (b) that includes popular literary forms such as hypertext fiction (Stuart Moulthrop's *Victory Garden*), interactive characters (*Eliza/Doctor*), and interactive fiction (Marc Blank and Dave Lebling's *Zork*). We might call these "environmentally" and "audience" interactive digital literature. It should be noted, of course, that this distinction (unlike others above) is not exclusive. *The Impermanence Agent* (by Noah Wardrip-Fruin, Brion Moss, Adam Chapman, and Duane Whitehurst) is an example of digital literature that is both environmentally and audience interactive.[16]

There is the potential for these sorts of distinctions to be of use in our conversations about digital literature, especially in combination (or tension) with existing groupings based on perceived genre. They help us name more precisely, for example, how the computationally variable email messages delivered by Rob Bevan and Tim Wright's *Online Caroline* (which are part of a larger audience interactive system) differ from those of most email narratives (which are generally computationally fixed) though the messages themselves are not interactive in either case.[17] These distinctions may also help us understand the relationship between the body of work in digital literature and in the broader digital arts—as well as the relationship between digital literature and computational systems more generally. Finally, we may also begin to use terms such as "random" or "interactive" more specifically.

But I believe there remains more I need to learn to read, in order to read digital literature.

Context

5.1 Computation as Context

At this point, in order to continue to broaden our view of digital literature, we need to begin to consider works that I still, frankly, find a bit puzzling. Here is an example of the kind of question that puzzles me: How do we understand the difference between an email narrative such as *Blue Company* and Bram Stoker's *Dracula*? Both, after all, are epistolary stories. Neither is interactive or otherwise computationally variable. But *Blue Company*'s letters originally arrived in one's email reader, with appropriate datestamps, and the timing of their arrival determined the possible timings of one's reading experience. Does this mean that there would be no difference between them if *Dracula* were separated into pieces and sent by post, receiving appropriate postmarks? No, not quite. When I say that digital literature requires digital computation, understanding computation required as context (e.g., the email reader as necessary context) is one of the challenges.

And this isn't simply for email narratives, or blog fictions, or other obviously networked forms (though Jill Walker Rettberg has begun interesting work on these and related forms under the term "distributed narrative"). As discussed earlier, works such as *Dakota* could have been created as traditional animations, and distributed on film. The dramatic growth of work in such forms isn't, however, simply an outgrowth of the availability of computer animation tools. There is something about the network, and about the growth of network culture (especially forums for posting, finding, sharing, and rating works—from sites specific to particular animation aesthetics through the teeming heterogeneity of *YouTube*) that has been important to the development of this work. And something about the ability to browse for and view this work in a web browser, using the same machine used for work, during any brief break from work. Here it seems likely that those of us studying digital literature could learn from work undertaken by Rettberg's colleagues in the Association of Internet Researchers.

But we also have a tradition, in digital literature, of "artifactual" work—which presents itself as a collection of computer files or systems, rather than as a literary work. The operations of a piece such as John McDaid's *Uncle Buddy's Phantom Funhouse* (delivered as a box of digital and non-digital artifacts supposedly left behind by the reader's recently deceased uncle) have nothing to do with the network. As McDaid explains:

> To be precise, in artifactual hypertext, the narrator disappears into the interface, with the logic of the hypertext becoming the "narration." Which is why, in cases where you are creating a fictional narrator who might be given to puzzles or games, such devices can be appropriate. But only within, and as aspects of, that narrating interface. (10)

Bill Bly, author of the artifactual *We Descend,* joins McDaid in suggesting that such work is not unique to digital forms, citing Milorad Pavic's *Dictionary of the Khazars* (36). Just as we may learn about how to interpret processes through the work of those who study artists such as Cage, we may learn techniques for approaching artifactual work from those who have interpreted print texts such as Pavic's or Ursula LeGuin's *Always Coming Home.* At the same time, the fact that some digital artifactual literature is interactive—that, for example, as McDaid suggests, there can be puzzles that, when solved, alter the operations of the work's processes—points to the limits of comparisons, as does the issue of context noted above in relation to email narratives.

5.2 Computation in Context

Just as the WIMP (Windows, Icons, Menus, and Pointers) interfaces of modern computers provide a context for much digital literature, it is also important to note that other digital literature embeds its computation and data in utterly different contexts. Perhaps it will help clarify the issues if we ask ourselves another puzzling question, such as one first posed to me by Roberto Simanowski: How do we understand the difference between Guillaume Apollinaire's "Il Pleut" and Camille Utterback and Romy Achituv's *Text Rain*? Apollinaire's poem is made up of letters falling down the page like rain. Utterback and Achituv's installation takes a video image of the audience standing before it and projects that image on the wall in front of the audience, with the addition (in the video scene) of the letters of a poem falling down like rain and resting on the bodies of their readers. Obviously, one difference is the passage of time in *Text Rain,* and another difference is that *Text Rain* is audience interactive (lifting up a hand on which letters rest causes them to be raised as well). But, at least as fundamentally, another difference is that *Text Rain* is situated in a physical space other than a printed page or a computer screen, in which the method of interaction is the movement of the readers' bodies (which are represented within the work itself). I would suggest that one way of conceptualizing this is through the idea of a work's *surface,* which gives the audience access to the results of its data and processes and through which any audience interaction occurs. The surfaces of "Il Pleut" and *Text Rain* are obviously radically divergent. Simanowski has begun to think through the issues we need to consider when interpreting digital literature of *Text Rain*'s sort from a literary perspective (cf. Simanowski), but the insights of disciplines such as performance studies will also be important as we investigate further.

There are a number of forms of digital literature for which space and the body are obviously essential to our consideration—including installation art such as *Text Rain* or Bill Seaman's literary installations, locative fictions such as *Itinerant*, dance and technology pieces such as Jamie Jewett and Thalia Field's *Rest/Less*, and literary virtual reality such as *Screen* (by Noah Wardrip-Fruin, Josh Carroll, Robert Coover, Shawn Greenlee, Andrew McClain, and Ben Shine).[18] But, as N. Katherine Hayles reminds us, we're not necessarily well served by ignoring the reader's body when interpreting other works of digital literature (Hayles, "Response"). It is worth attending to the ways that our bodies become trained in the unusual WIMP mousing behavior required to engage the surface of Talan Memmott's *Lexia to Perplexia* or in the combinations and timings of game console controller manipulations required to move through Jordan Mechner and Ubisoft's *Prince of Persia: The Sands of Time.*

Of course there is also another kind of context—social context—that we have not yet discussed. Another mildly puzzling question might help bring some of these issues to the fore: How do we understand the difference between an interactive fiction such as *Zork* and a MUD or MOO? An interactive fiction is a textually-described world which one can move through by typing commands: investigating spaces, acquiring objects, and interacting with characters. MUDs and MOOs share all these characteristics with interactive fictions—the primary difference, for a first time visitor, is that the characters in the space are often real people (other visitors, experienced participants, and even those involved in constructing the world). Torill Mortensen is one of the writers who has been thinking seriously about the pleasures of experiencing these textual worlds with other players, as they are shaped through time by the actions of other players (cf. Mortensen). In a related vein, T.L. Taylor has been writing about graphical environments such as *EverQuest* in a manner that foregrounds how interactions within the simulated world are shaped by networks of relation "outside" of it (cf. Taylor and Jakobsson). Work of this sort is necessary if we are going to understand player experiences, and the context in which the performative narrative interventions of "event teams" take place in worlds such as *EverQuest* and more recent massively-multiplayer games, as well as related forms such as alternate reality games (which, following Elan Lee and Sean Stewart's foundational project *The Beast*, often involve elaborate plots and puzzles, hundreds of documents, and thousands of simultaneous reader/players in communication).

But there is much more we might consider as social context. Take, for example, the case of *Eliza/Doctor*. Our interpretations of this work are likely to be a bit odd if we interact with it now in a graphical window (e.g., as a Java applet in a web browser) and never consider its original context. As Nick Montfort has pointed out, the project was developed on a system that not only wasn't a graphical screen—it wasn't a screen at all. Rather, the system's textual interactions (and the reader's replies) were printed on a continuous ream of paper fed through a teletype, a surface nearly forgotten in our contemporary world of digital media ("Continuous Paper"). I draw attention to this not simply in order to point out that a work's surface can vary as it is experienced in different computational environments. More importantly, *Eliza/Doctor*'s typewritten interaction was taking place in an environment in which people communicated with each other through the very same textual medium (much as many people communicate via instant messenger clients today). This is what made possible the famous story of the Bolt Beranek and Newman manager conducting an increasingly exasperated conversation with *Eliza/Doctor*, believing himself to be communicating with a subordinate. As Janet Murray has pointed out, the spread of this story (in several variations, some certainly apocryphal) mirrors that of the Paris audience that supposedly fled the theatre when the Lumière Brothers' film of an approaching train was first shown (65-66). It points to our anxiety that the representational power of a new medium might cause us to mistake its products for reality. In the world of BBS culture, where I first experienced *Eliza/Doctor*, it remained in a context of predominantly textual software experiences mixed with human-to-human textual communication that allowed it to retain much of its original impact.[19] But to interact with *Eliza/Doctor* now, even if running as a bot on an instant-messaging network, is to read the work in a context quite substantially different from that in which it was created and first experienced.

6 A Five Element Model

6.1 Expanding Aarseth's Models

Espen Aarseth's *Cybertext: Perspectives on Ergodic Literature* is one of the most important books for those interested in digital literature to consider. The two neologisms in the title are worth defining. "Ergodic literature" is literature in which "nontrivial effort is required to allow the reader to traverse the text" (1). This can include works, such as James Meehan's *Tale-Spin*, that have both interactive and non-interactive modes of operation—-or even works that involve simple choices of eye movement, such as poems of Apollinaire's that involve reader exploration on the page (or Egyptian wall inscriptions connected in two dimensions). As for the other neologism, Aarseth writes on page 3, "A cybertext is a machine for the production of variety of expression." This includes works in which texts can be added by the reader, or in which texts can be generated differently from fixed initial materials, or in which connections between texts can change in different states of the work. It explicitly excludes, however, statically-linked hypertexts such as Stuart Moulthrop's *Victory Garden* (cf. 75 and fig. 3.2).

Aarseth's book provides two models, each with three parts, that have been widely used by those writing about digital literature (though neither of his models is in any way limited to digital literature). The first, in chapter 1, is of the "textual machine" as composed of verbal sign, medium, and operator. As Aarseth writes:

> As the *cyber* prefix indicates, the text is seen as a machine—not metaphorically but as a mechanical device for the production and consumption of verbal signs. Just as a film is useless without a projector and a screen, so a text must consist of a material medium as well as a collection of words. The machine, of course, is not complete without a third party, the (human) operator, and it is within this triad that the text takes place. (21)

Aarseth's second model appears later. He writes:

> A text, then, is any object with the primary function to relay verbal information. Two observations follow from this definition: (1) a text cannot operate independently of some material medium, and this influences its behavior, and (2) a text is not equal to the information it transmits. *Information* is here understood as a string of signs, which may (but does not have to) make sense to a given observer. It is useful to distinguish between strings as they appear to readers and strings as they exist in the text, since these may not always be the same. For want of better terms, I call the former *scriptons* and the latter *textons*. Their names are not important, but the difference between them is. In a book such as Raymond Queneau's sonnet machine *Cent mille milliards de poèms,* where the user folds lines in the book to "compose" sonnets, there are only 140 textons, but these combine into 100,000,000,000,000 possible scriptons. In addition to textons and scriptons, a text consists of what I call a traversal function—the mechanism by which scriptons are revealed or generated from textons and presented to the user of the text. (62)

Aarseth then proposes seven variables that "allow us to describe any text according to their mode of traversal." These are: dynamics, determinability, transiency, perspective, access, linking, and user function. It is difficult to overstate the importance of Aarseth's contributions to thinking about digital, ergodic, and cybertext literature. But I have also found, in thinking through what I believe I need to interpret in order to read digital literature, that for my purposes Aarseth's models need some expansion. For example, works of digital literature carry out many processes—such as those determining the simulated emotional state of a virtual character—that are important to their literary functions but are not traversal functions for revealing or generating textons from scriptons (or can only be considered as such quite circuitously). Somewhat differently, Aarseth's model of the textual machine (often represented as an equilateral triangle) divides the work up into a "material medium" and "a collection of words." While we can easily expand Aarseth's collection of words to data of many types, a work's processes are as much a matter of authorial creation and selection as its data, and can hold steady while its surface varies.

Given this, I find myself more comfortable using the five-part model presented here, rather than either of Aarseth's three-part models, as my starting point for reading digital literature. I should emphasize, however, that I view my work here as an expansion of, rather than a rejection of, Aarseth's work. All that said, I would summarize the five-part model presented here as follows:

> *Data.* This includes text, images, sound files, specifications of story grammars, declarative information about fictional worlds, tables of statistics about word frequencies, and so on. It also includes instructions to the reader (who may also be an interactor), including those that specify processes to be carried out by the reader.
>
> *Processes.* These are processes actually carried out by the work, and are central to many efforts in the field (especially those proceeding from a computer science perspective). As Chris Crawford puts it: "Processing data is the very essence of what a computer does." Nevertheless, processes are optional for digital literature (e.g., many email narratives carry out no processes within the work) as well as for ergodic literature and cybertext (in which all the effort and calculation may be on the reader's part).
>
> *Interaction.* This is change to the state of the work, for which the work was designed, that comes from outside the work. For example, when a reader reconfigures a combinatory text (rather than this being performed by the work's processes) this is interaction. Similarly, when the work's processes accept input from outside the work—whether from the audi-

ence or other sources. This is a feature of many popular genres of digital literature, but it is again optional for digital literature and cybertext (e.g., *Tale-Spin* falls into both categories even when not run interactively) and for ergodic literature as well (given that the page exploration involved in reading Apollinaire's poems qualifies them as ergodic). However, it's important to note that cybertext requires calculation somewhere in the production of scriptons—either via processes or interaction.

Surface. The surface of a work of digital literature is what the audience experiences: the output of the processes operating on the data, in the context of the physical hardware and setting, through which any audience interaction takes place. No work that reaches an audience can do so without a surface, but some works are more tied to particular surfaces than others (e.g., installation works), and some (e.g., email narratives) make audience selections (e.g., one's chosen email reader) a determining part of their context.

Context. Once there is a work and an audience, there is always context—so this isn't optional. Context is important for interpreting any work, but digital literature calls us to consider types of context (e.g., intra-audience communication and relationships in an MMO fiction) that print-based literature has had to confront less often.

These are, of course, far from rigid categories, as well as deeply dependent on each other. To take an example, supporting particular interactions is, of course, dependent on using a surface to influence appropriate processes and data. Or, to look at things more formally, as Crawford points out, "Experienced programmers know that data can often be substituted for process. Many algorithms can be replaced by tables of data. . . . Because of this, many programmers see process and data as interchangeable." But I think such arguments generally grow from approaching the issues at the level of minutia, rather than an attempt to think about what is important in interpreting digital literature. As Crawford says of the case he mentions, "This misconception arises from applying low-level considerations to the higher levels of software design."

6.2 Reading Processes

In the end, however, it isn't important how one divides up the elements of digital literature, or how one defines digital literature, except in how it informs analysis and creation.[20] I think it is important to distinguish process and surface, rather than collapse both into the "medium," in part because I believe that a major next step for our field is to begin to interpret processes. As I argue above in the context of Strachey's work, there are works of digital literature we simply can't understand without investigating their processes. Further, there are at least three ways that processes recommend themselves to our attention. First, they are a powerful means of crafting media experiences. Second, they express relationships, through their designs, with schools of thought, histories of practice, and other configurations important to interpretation. Third, because the processes of digital literature often operate in terms of concepts of humanity and the world (e.g., language, character motivation) they can be seen as miniature, operationalized philosophies of these concepts. These philosophies can be worth investigating in their own rights, as well as in how they shape the audience experience—both during "normal" operation and in situations of breakdown (which are not uncommon in complex digital systems). My book *Expressive Processing* is an early attempt at reading processes in these ways. I propose the elements suggested here, in part, because I believe that more attention to process, and to each of the other elements as identified here, will enrich the field. As investigation moves forward, I hope that new organizations, focusing on other elements, will naturally replace this one.

Notes

1 To be precise, "computer" was a job title. As N. Katherine Hayles writes in the prologue to *My Mother Was a Computer*, "in the 1930s and 1940s, people who were employed to do calculations—and it was predominantly women who performed this clerical labor—were called 'computers.' Anne Balsamo references this terminology when she begins one of the chapters in her book *Technologies of the Gendered Body* with the line I have appropriated for my title: 'My mother was a computer.' Balsamo's mother actually did work as a computer, and she uses this bit of family history to launch a meditation on the gender implications of information technologies" (1).

2 The design of the Analytical Engine called for it to be programmed via punched cards such as those used for automated looms, making it possible for Babbage's collaborator Ada Lovelace to be called by some the first programmer of a universal computer, even though the Analytical Engine was never constructed. Lev Manovich, in *The Language of New Media*, has commented on this connection between looms and computers, writing: "Thus a programmed machine was already synthesizing images even before it was put to processing numbers. The connection between the Jacquard loom and the Analytical Engine is not something historians of computers make much of, since for them computer image synthesis represents

just one application of the modern digital computer among thousands of others, but for a historian of new media, it is full of significance" (22).

3 In contrast, many early 20th century computers used analog, continuous representations—such as varying electrical currents or mechanisms that turned at varying speed. These analog computers could perform some tasks very quickly. For example, adding two quantities represented by electrical currents could be accomplished simply by allowing flow onto particular wires, rather than by actually establishing the two values and numerically calculating their sum. However, because of the lack of discrete states, analog computers were inflexible in their orders of precision and prone to noise-induced errors.

4 Many other projects and incremental advances took place, and especially notable of these was the University of Pennsylvania ENIAC (believed, while the *Colossus* was still secret, to have been the first fully functioning electronic digital computer). A 1945 report of future design plans-—based on insights from ENIAC designers J. Presper Eckert and John Mauchly, working together with John von Neumann—was very influential on the design of future stored program digital computers (leading to the perhaps inappropriate name "von Neumann architecture" for such systems).

5 Similar efforts include the University of Cambridge EDSAC (1949), the University of Pennsylvania EDVAC (1951), the MIT *Whirlwind I* (1949), and others.

6 It was only a few years later that Turing committed suicide-—after arrest and conviction for homosexual activities, followed by a sentence of hormone injections that caused him to grow breasts.

7 However, David Durand's research in the Oxford Bodleian Library has unearthed the program's complete grammar and vocabulary, which included "love," "loves," "loving," "lovingly," "lovesick," and "lovable."

8 For more on the love letter generator, cf. "Digital Media Archaeology: Interpreting Computational Processes" (Wardrip-Fruin, forthcoming).

9 *Dakota,* according to its authors "is based on a close reading of Ezra Pound's Cantos I and first part of II" (Swiss)—but in decidedly modern language. For example, Pound's opening line "And then went down to the ship" becomes "Fucking waltzed out to the car." Stark black text about driving and drinking, guns and gangbangs and Elvis, appears on a white background in time to Art Blakey's jazz drumming, eventually accelerating into near-illegibility.

10 *Blue Company* is an email novel that was performed in 2001 and 2002, with the current news influencing how messages were sent. As succinctly described in the Electronic Literature Organization's "showcase" entry for the piece, "a 'new economy' worker who is sent back in time to the early renaissance tells the story of his corporate team, Blue Company, and their curious work as he writes e-mails on an illicit laptop to his inamorata." *The Daughters of Freya* is also delivered as a series of fictional email messages, but even in its form is quite different from *Blue Company.* For example, the work is not performed at particular times, but always available to new readers, who receive their first message a few hours after signing up at the project's website. Also, while *Blue Company* maintains a close correspondence between the messages sent by the characters and those received by readers, *The Daughters of Freya* often includes messages from multiple characters in a single email received by readers, with datestamps driven by the story's timeline rather than the time of reading. And the two stories are also quite different, with that of *The Daughters of Freya* focusing on a reporter investigating a California sex cult and, eventually, a murder.

11 Natural Language Generation is an area of computer science research that focuses on the production of "natural language" (e.g., English). Tools employed range from formally-described grammars to word frequency statistics. A small number of researchers focus specifically on the generation of text with a particular stylistic approach or other literary parameters.

12 Cage is notable for using "chance operations" in his compositions, and also for scoring processes (such as the tuning of radios) that would reveal sound data that could not be determined in advance. Fluxus works include those that are purely process specifications (lists of instructions) as well as performances and other works built on process models. Augusto Boal's dramatic work is largely in the construction and use of his participatory theatre techniques (such as "forum theatre") with new actor/writers and "spect-actor" audiences.

13 Though, in common usage, the term "Turing test" usually drops the imitation game and gender aspects of Turing's description—focusing instead on whether a human judge believes a textual conversant to be human or machine. Artworks such as Mark Marino's *Barthes's Bachelorette* and Greg Garvey's *Genderbender 1.0* playfully recover some of these aspects.

14 This may be due, in part, to the descriptive vocabulary developed for hypertext systems. While Michael Joyce's distinction between "exploratory" and "constructive" hypertexts is certainly useful when trying to explain different types of hypertext systems and user positions, it's important

to remember that hypertexts that function in what Joyce would call an "exploratory" manner may not have exploration as their primary user experience.

15 *Itinerant* (2005) is a site-specific sound installation in Boston, Massachusetts. It invites people to take a walk through Boston Common and surrounding neighborhoods to experience an interactive sound work delivered via handheld computer and driven by GPS satellite information. During a walk which may last for more than two hours, visitors explore the area, finding fragments of a personal narrative of family and displacement, interspersed with passages from Mary Shelley's *Frankenstein*—the classic tale of a technoscientific monster and the family love he witnesses voyeuristically, but cannot share. It is an exploration-based narrative, but there are no links to click. "The Carl Comics," on the other hand, use links for purposes other than exploration. For example, *Original Recipe Carl* (1998) tells the story of Carl's death in an auto accident. Clicking links allows the reader to expand or contract the number of panels it takes from when Carl promises not to drink and drive until we end at his tombstone—from two panels to fifty-two panels. In essence, links change the level of detail of the story, making it like a comics version of one of Ted Nelson's original non-chunk hypertext concepts: "stretchtext." *Choose Your Own Carl* (1998-2001), on the other hand, is a crossword-style comic (branching and recombining) on the same subject, which is composed of frames drawn based on the suggestions of more than a thousand readers. Here, link-clicking operates to reveal the original suggestions considered for each frame. The result, as McCloud puts it, is a "Fully Interactive, Multiple Path, Reader-Written, Death-Obsessed Comics Extravaganza."

16 *The Impermanence Agent* (1998-2002) is not interactive in the sense that the audience can, say, click on the work. This piece launches a small browser window and tells a story of documents preserved and lost, of impermanence, within it. While this story is being told, the work is also monitoring the reader's web browsing of other sites. Parts of sentences and images from the reader's browsing are progressively collaged into the story, using a variety of techniques. This results in a different experience for every reader---one which is environmentally interactive in that it draws its material primarily from websites created without the work in mind, but is audience interactive in that readers can choose to alter their browsing habits in order to provide the work with different material to consider for collage into the story (cf. Wardrip-Fruin and Moss).

17 *Online Caroline* (2000, 2001) not only sends the reader email messages—it expects a response. The responses don't come by writing email, but by visiting the website of the reader's online friend, Caroline. At the website readers communicate with Caroline via a simulated webcam, enter details about themselves via web forms, and experience the unfolding of a 24-part drama. Each email sent to a reader is a fixed block of text, but these texts are customized based on what is known about the reader from website visits (e.g., whether the reader has children). Also, the sequence of messages is not fixed. If a reader goes too long without visiting the website, after the receipt of one of Caroline's email messages, the character will begin to send reminders and eventually break off the "relationship" with a message that includes the words: "I won't mail you any more. I'll assume you're away, or busy . . . or maybe you're just fed up with me."

18 *Rest/Less* brings poetry together with dance, music, and technology. A collection of grid-shaped poems by Field become the space over which five dancers choreographed by Jewett move—triggering spoken language, bells, wind, and video images of the handwritten poems. The performance system, developed by Jewett, does not require sensors or tracking aids to be placed on the dancers' bodies, leaving them free to interact lyrically with the grid made visible to the audience on the floor of the performance space. *Screen* was created in Brown University's "Cave," a room-sized virtual reality display. It begins as a reading and listening experience. Memory texts appear on the Cave's walls, surrounding the reader. Then words begin to come loose. The reader finds she can knock them back with her hand, but peeling increases steadily.

19 BBSes, or Bulletin Board Systems, were computers that accepted connections from other computers over regular phone lines, using modems. Often run by individuals as a community service (though there were also commercial BBSes) these machines usually had a small number of dedicated phone lines that allowed users to upload and download files, take part in asynchronous discussions on "bulletin boards," make moves in tum-based games, exchange real-time messages with the users currently connected to the other phone lines, and interact with programs like *Eliza/Doctor.* A vibrant culture in the 1980s, they disappeared almost overnight as public access to the Internet expanded.

20 In fact, the elements presented in this essay are ones selected for the purpose of my argument here—and I have used others on different occassions.

Works Cited

Aarseth, Espen J. "Nonlinearity and Literary Theory." *Hyper/Text/Theory*. Ed. George P. Landow. Baltimore: Johns Hopkins, 1994.

———. *Cybertext: Perspectives on Ergodic Literature*. Baltimore: Johns Hopkins UP, 1997.

Bates, Joseph. "The role of emotion in believable agents." *Communications of the ACM* 37.7 (1994).

Betcherman, Michael, and David Diamond. *The Daughters of Freya*. 2005. 10 Aug. 2009 <http://www.emailmystery.com/dof/index.php>.

Bevan, Rob, and Tim Wright. *Online Caroline*. 2001. 10 Aug. 2009 <http://www.onlinecaroline.com>.Bly, Bill. "Learn navigation: doing without the narrator in artifactual fiction." *SIGWEB Newsletter* 9.1 (2000): 34-37.

Bringsjord, Selmer, and David A. Ferrucci. *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, a Storytelling Machine*. Lawrence Erlbaum Associates, 2000.

Campbell-Kelly, Martin. "Christopher Strachey 1916-1975: A Biographical Note." *Annals of the History of Computing* 7.1 (1985).

Crawford, Chris. "Process Intensity." *Journal of Computer Game Design* 1.5 (1987).

Douglass, Jeremy. "Machine Writing and the Turing Test." Presentation in Alan Liu's Hyperliterature seminar, University of California, Santa Barbara. 2000. 10 Aug. 2009 <http://www.english.ucsb.edu/grad/student-pages/jdouglass/coursework/hyperliterature/turing/>.

Field, Thalia, and Jamie Jewett. *Rest/Less*. 2005. Performance.

Hayles, N. Katherine. "Response to Simon Penny." Riposte. *Electronic Book Review* (2004). 10 Aug. 2009 <http://www.electronicbookreview.com/thread/firstperson/haylesr1>.

———. *My Mother Was a Computer: Digital Subjects and Literary Texts*. Chicago: U of Chicago P, 2005.

Hershman, Lynn. *Synthia*. 2002. 10 Aug. 2009 <http://www.lynnhershman.com/synthia/>.

Hillis, W. Daniel. *Pattern on the Stone: The Simple Ideas That Make Computers Work*. New York: Basic Books, 1999.

Hodges, Andrew. *Alan Turing: The Enigma*. New York: Walker, 2000.

Joyce, Michael T. "Siren Shapes: Exploratory and Constructive Hypertexts." *Academic Computing* 3 (1988).

Klima, John. *Earth*. 2001. Installation.

Lang, R. Raymond. "A formal model for simple narratives." *Narrative Intelligence: Papers from the 1999 AAAI Fall Symposium,* Technical Report FS-99-01. Ed. Michael Mateas and Phoebe Sengers. AAAI Press, 1997.

Lebling, Dave, and Marc Blank. *Zork*. Computer Game. 1977.

Guin, le Ursula K. *Always Coming Home*. Berkeley: U of California P, 2001.

Manovich, Lev. *The Language of New Media*. Cambridge, MA: MIT P, 2001.

McCloud, Scott. *The Carl Comics*. 1998-2001. 10 Aug. 2009 <http://www.scottmccloud.com/1-webcomics/carl/index.html>.

McDaid, John. "Luddism, SF, and the Aesthetics of Electronic Fiction." *New York Review of Science Fiction* 69 (1994): 1; 8-11.

Meriwether, James B. "Notes on the Textual History of The Sound and the Fury." *Papers of the Bibliographical Society of America* 56 (1962): 285-316.

Montfort, Nick. "Literary Games." *Poems that Go* 14 (2003). <http://www.poemsthatgo.com/gallery/fall2003/print_article_games.htm>.

———. "Continuous Paper: The Early Materiality and Workings of Electronic Literature." *Modern Language Association Conference*. 2004. <http://www.nickm.com/writing/essays/continuous_paper_mla.html>.

Mortensen, Torill. "Flow, Seduction and Mutual Pleasure." *Other Players Conference Proceedings*. Center for Computer Game Research, ITU Copenhagen, 2004.

Moulthrop, Stuart. *Victory Garden*. CD-ROM. Cambridge, MA: Eastgate, 1991.

Murray, Janet H. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. New York: Free Press, 1997.

Paul, Christiane. *Digital Art*. London: Thames and Hudson, 2003.

Queneau, Raymond. *Cent Mille Milliards de Poèmes*. Paris: Gallimard, 1962.

Rueb, Teri. *Itinerant*. Installation. 2005.

Simanowski, Roberto. "Holopoetry, Biopoetry and Digital Literatures: Close Reading and Terminological Debates." *The Aesthetics of Net Literature: Writing, Reading and Playing in Programmable Media*. Ed. Peter Gendolla and Jörgen Schäfer. Bielefeld: Transcript, 2007. 43-66.

Strachey, Christopher. "The 'Thinking' Machine." *Encounter* 3.4 (1954): 25-31.

Swiss, Thom. "Distance, Homelessness, Anonymity, and Insignificance: An Interview with Young-Hae Chang Heavy Industries." *The Iowa Review Web*.

2002. Aug. 2009 <http://www.uiowa.edu/~iareview/tirweb/feature/younghae/interview.html>.

Taylor, T. L., and Mikael Jakobsson. "The Sopranos Meets EverQuest: Socialization Processes in Massively Multiuser Games." *Digital Arts and Culture::2003::Streaming Wor(l)ds.* Melbourne, 2003.

Alan M. "On computable numbers with an application to the Entscheidungsproblem." *Proceedings of the London Mathematical Society* 2.42 (1936).

"Computing Machinery and Intelligence." *Mind: A Quarterly Review of Psychology and Philosophy* 59.236 (1950).

Young-Hae Chang Heavy Industries. *Dakota.* 2002. 10 Aug. 2009 <http://www.yhchang.com/DAKOTA.html>.

Walker-Rettberg, Jill. "Distributed Narrative: Telling Stories Across Networks." *The 2005 Association of Internet Researchers Annual.* Ed. Mia Consalvo, Jeremy Hunsinger, and Nancy Baym. New York: Peter Lang, 2006.

Wardrip-Fruin, Noah. "What Hypertext Is." *Proceedings of the Fifteenth ACM Conference on Hypertext and Hypermedia.* ACM, 2004.

———. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies.* Cambridge, MA: MIT P, 2009.

———. "Digital Media Archaeology: Interpreting Computational Processes." *Media Archaeologies.* Ed. Erkki Huhtamo and Jussi Parikka. Berkeley: U of California P. Forthcoming.

Wardrip-Fruin, Noah, and Nick Montfort, eds. *The New Media Reader.* Cambridge, MA: MIT P, 2003.

Wardrip-Fruin, Noah, and Brion Moss. "The Impermanence Agent: Project and Context." *Cybertext Yearbook 2001.* Ed. Markku Eskelinen and Raine Koskimaa. Jyväskylä: Research Centre for Contemporary Culture, 2002. 14-59.

Wardrip-Fruin, Adam Chapman, Brion Moss, and Duane Whitehurst. *The Impermanence Agent.* 1999. 10 Aug. 2009 <http://www.impermanenceagent.com/agent/>.

Wardrip-Fruin et al. *Screen.* 2002. 10. Aug. 2009 <http://www.noahwf.com/screen/index.html>.

Wegner, Peter. "Why Interaction is More Powerful than Algorithms." *Communications of the ACM* 40.5 (1997): 80-91.

Weizenbaum, Joseph. *Computer Power and Human Reason: From Judgment to Calculation.* San Francisco: W.H. Freeman, 1976.

Wittig, Rob. *Blue Company.* 2002. 10 Aug. 2009 <http://www.robwit.net/bluecompany2002/>.