Modes of Dialogue and Editing in Digital Literature

Jerome Fletcher

1. INTRODUCTION

This paper is designed to make a contribution to the discussion around the topic of »Writing as dialogue« but to do it from the perspective of what is still something of a niche writing practice, i.e. digital literature. By taking this slightly oblique perspective, the hope is that we may also provide some insight into the role of dialogue in more mainstream literary forms and modes.

I will begin by uncovering and exploring multiple processes of dialogue and editing in relation to creative composition in digital literature and will argue that dialogue is not an adjunct to digital writing, it sits at its very heart. In its weaker form this is an argument that can be applied more or less to any text, digital or print. Writing is a call to which reading is a response. However, I would like to make a case for digital text/literature as a paradigm for this view of writing as dialogic, of writing-as-rewriting. Inevitably, the digital writer can only ever produce a text of radical incompleteness. Moreover, I aim to show that in order to make a digital text perform, a series of »dialogic relationships« between various elements of the digital apparatus is required (machine, codes, interface, etc.), and often an »editorial intervention« on the part of the digital reader/user. This paper will also argue that given the performativity of digital text, it exists as an integrated text for no longer than the duration of its performance. Beyond that, it can only ever constitute a dispersed potential. In other words, where digital literature is concerned, writing will by definition be processual, dialogic and, primarily, editorial.

I am aware that in exploring the concepts of editing and dialogue in the context of *digital* literature, I am using these terms in a more expanded sense than they might be used in *print* literature and as is suggested in the title of the book. First and foremost, the term »dialogue« will encompass interaction with a machine and within a machine, as well as between machines. Here we are dealing with a form of dialogic activity that implies, but does not necessarily re-

quire, the physical presence of another being. As such, this approach separates itself from the primary notion of »literary« dialogue as it may be understood in a »mentor-student exchange« or in an »editor-writer discussion«. At the same time, digital modes of dialogue may involve the presence of two or more co-writers or text-editors, who do not necessarily engage in a verbal exchange but who intervene in a text within a digital space or setting, without meeting or speaking with one another. The present discussion of the interaction between code and text and of specific digital projects will further specify how dialogue is to be defined and understood in this particular context. It will show how digital dialogic practices are akin to collaborative writing processes, which also include a dialogue between texts.

As for the use of the term »editing«[1] in the context of digital writing, it implies that other (co-)writers (whether code- or text-writers) are involved in the online outcome of the text (or code). A »digital editor« thus may not have written the source text but intervenes in and changes it, sometimes radically, as it is continually re-published or re-performed in its digital environment. The scope of the intervention will vary and, in some projects, this intervention will be akin to »co-writing« rather than to »editing« in its more conventional meaning. »Editing« implies a secondary intervention in the text – the notion of original text fluctuates here too, since it may become unrecognisable through various layers of modifications. In digital writing processes, boundaries between writing, co-writing and editing remain porous; collaborative practices are very often at the core of the production of literary texts, and the »editor« should be thought of as a subject position rather than as an identifiable individual. Digital »editing processes« will hence be specifically defined in each of the projects discussed further in this contribution.

While not wanting to get into a lengthy discussion on the exact nature of digital literature, it may be worth making a few preliminary remarks. It is still the case that there is no broadly accepted consensus on what constitutes »digital literature«. (But then »literature« itself is still a contested term.) Certain attempts at definition are useful, like John Cayley's »networked programmable literature«, as is the Electronic Literature Organisation's list of digital literature forms: Hypertext fiction, network fiction, interactive fiction, locative narrative, installation pieces, »codework/code poetry«, poetry generators, and the Flash

---

1 | The English notion of »editing« does not correspond to the notion of *lektorieren* discussed in other German contributions of the present book. In the English-speaking context, the function of the editor in conventional literary publications (printed or digital books) may vary according to the situation; thus an editor may »proofread« a text, intervening on its surface; »editing« may also imply a deeper »revision process« prior to the publication of a text, intervening – most often in dialogue with the author – on matters of language, style and structure, etc.

poem,[2] among others. The sine qua non of digital literature is of course the computer, both for its composition and for its access, display and dissemination. All the time we should bear in mind that, strictly speaking, the digital constitutes only a small element of a computer. Much of what makes up the machine is very analog. One consequence that flows from this is that digital literature entails at least two forms of writing: literary/creative composition and coding. The objection may be made that the latter is irrelevant to any discussion about literature proper. However, I would argue simply that if a computer is not involved in the composition, display and dissemination of a digital text work, then it has no claim as a work of digital literature. It is quite possible to use a computer to compose a piece of print-based writing, but this does not make the outcome – the printed text – a piece of digital literature. By the same token it is quite possible to engage in purely process-driven, mechanical writing (as Oulipo does, for example) without creating a piece of digital text. It was to separate works of digital literature from digitally-produced print texts that Katherine Hayles coined the term »digital born«.[3] This is not a trivial concern, in that if the computer is a necessary condition of digital literature, then in order to give a full analysis of the process of digital writing, we have to take into account the entire digital apparatus (hardware and software) and its claims to creativity. As stated above, it is the dialogic relationship between these various elements of the whole digital apparatus and across connected apparatuses that is the focus of this paper.

2. WRITING CODE AS DIALOGUE

Given the prerequisite of the machine as an integral part of the creative process, one thing that can be stated from the outset is that even before any sort of imaginative/literary writing takes place in the digital environment, a number of different writing processes will already have occurred within the machine itself. These include the writing of the operating code as well as the programme software used to create and display the digital text work. Some theorists such as Florian Cramer have argued that this aspect of writing does not »count« as a form of writing because it is too instrumental, entailing nothing but the composition of impersonal instructions – that the writing of code lacks the social dimension or engagement of natural languages. As he points out:

[C]omputer control language is language that executes. As with magical and speculative concepts of language, the word automatically performs the operation. Yet this is not

---

2 | See https://eliterature.org/pad/elp.html#preface

3 | Hayles, N. Katherine: »What is e-literature?«, https://eliterature.org/pad/elp.html#preface

to be mixed up with what linguistics calls a »performative« or »perlocutionary« speech act.[4]

Indeed, the operations of the machine have no a priori social meaning. Software engineer Ellen Ullman concurs: »[A] computer program has only one meaning: what it does. It isn't a text for an academic to read. Its entire meaning is its function.«[5] Of course, the notion of meaning rooted in function is at the heart of a Wittgensteinian theory of language. However, Cramer and Ullman's notions of (non-)performativity refer to code as a finished product, not as an ongoing writerly process. Against this view I would cite Scott Dexter's article, »Towards a Poetics of Code«, and Geoff Cox's *Speaking Code*, both of which consider coding as a social practice.

In »Towards a Poetics of Code«, Dexter lays out his view of code as a profoundly cultural practice, by focusing not on software-in-execution but software-in-creation, or source code. Dexter's argument is that source code is performative and dialogic in that it requires an audience which effectively acts as a »critical friend« or editor.[6] This is particularly true of free software source code, which is developed and put out in the public domain free of corporate control. It relies for increased efficiency and efficacy on feedback from a public audience, which is itself often involved in the process of rewriting or editing:

One of the many concerns of software engineering as an area of study and of practice is how to structure the developers on a project so that code is exposed to a critical audience as effectively yet efficiently as possible. In effect, the audience becomes a co-developer, or reviewer or bug reporter. One clear example is the practice of »pair

---

4 | Cramer, Florian: »Language«, in: Matthew Fuller (ed.), Software Studies: A Lexicon, Cambridge MA: MIT Press 2008, p. 185. Here Cramer is referring to the linguistic category of performatives proposed by ordinary language philosopher J. L. Austin in his book *How to do Things with Words* and later developed by his student, John Searle, in his *Speech Acts.* Both argue that much of language cannot be analysed in terms of truth and falsehood because it is not concerned with making statements, but with »doing things«, i.e. performing.

5 | Hayles, N. Katherine: My Mother was A Computer: Digital Subjects and Literary Texts, Chicago: University of Chicago Press 2005, p. 48.

6 | The purpose of editing code is often to improve efficiency, conciseness and increase performativity. At first sight, these appear to be very »unliterary« categories. However, there is an argument that editing a poetic text, for example, is a process of getting the language to do as much »work« as possible with the least amount of resource; i.e. a form of efficiency and increased performativity. Of course, there are also areas where code editing and literary editing have very different approaches. An obvious one would be in relation to ambiguity.

programming«, in which one developer writes code while the other looks on and strives to provide constructive critical feedback.[7]

It is possible to see this as not only an editorial process but also one of mentorship – in other words, pair programming is not simply a way of ensuring the efficacy of code, but takes on the outlines of a pedagogical relationship, where more experienced coders might instruct and develop the skills of less experienced practitioners.

There is a clear parallel here with the creative practice element of digital literature and with digital poetics in general. Particular examples of this are collaborative writing projects and any form of digital interactivity, which allows the audience to modify the work in significant ways. (These will be explored in greater detail in section 3 below.) This is true of the niche areas of digital literature as well as the broad mainstream writings of fan fiction, for example, where an established writer will specifically tailor a narrative to the ideas and desires suggested by their target audience.[8] A particularly telling example of this is the history of writing on the operating system Linux. As the »inventor« of Linux, in 1999 Linus Torvalds was the recipient of the prestigious Golden Nica prize awarded by Ars Electronica in Austria. The jury of the .net category prize recognised that Torvalds, however, was also »representing all of those, who have worked on this project [Linux] in past years and will be participating in it in the future [...]. It is also intended to spark a discussion about whether a source code itself can be an artwork.«[9] Needless to say, this phenomenon of an interactivity which is sought by the author raises intriguing questions about the nature of ownership and authorship. Who is doing the writing and who takes credit and/or responsibility for the writing? It is no longer simple, or maybe even possible, to separate the writing and editorial functions. It is not possible to think in terms of an ur-text complete in itself, which is then responded to and adapted for final publication. The nature of code is that it is a performative text. It is in a constant state of iteration. At any point the text is liable to updating, revision, rewriting. And as the code drives the digital »literary« output, that in turn is liable to constant revision, re-presentation, even disruption and possibly obsolescence when a particular software is no longer available.

---

7 | Dexter, Scott: »Towards a Poetics of Code«, http://www.academia.edu/2860624/Toward_a_Poetics_of_Code

8 | The whole area of fan fiction – at heart a dialogic and editorial enterprise between writer and readership – has a bearing on this topic, but lies outside the realm of digital literature proper. Although the writing of fan fiction largely takes place online, it is ultimately concerned with print-based texts. For this reason I acknowledge it but lay it to one side.

9 | S. Dexter: Poetics of Code.

Developing this discussion of code-as-dialogic-writing, the fullest treatment we have so far of code as something more than just instrumental is Geoff Cox's *Speaking Code*. Here he argues that code is closer to speech than to writing and relates code to a collective speech act, one that is doubly articulated around expression and function. He writes:

the combination of formal description and creative action, what might be referred to as double-coding, is well established in software arts practice. [...] This exemplifies the material aspects of code both on the functional and the expressive level, [...] involving both formal logic and expressive aspects, its constraints and excesses.[10]

For Cox, coding is above all an embodied practice:

The body is of course registered in the content, (in the codework itself) in the narrator's body (the comments and secondary notation), but also in the bodies of all those humans involved in the production process, including the reader's body.[11]

In this respect Cox understands code precisely as »performative« in the linguistic sense, which Cramer, cited above, disputes. Cox goes further, however. The subtitle of the book is »Coding as Aesthetic and Political Expression«. The latter claim is based on his view that not only do we make code speak, but code *speaks us*, pervading and formatting our actions and behaviours. Cox draws a parallel between this state of affairs and the way in which the act of interpellation (in the Althusserian sense) acts upon the body.[12] This adds an extra dimension to the question of writing and dialogue in digital literature. Coding is a two-way process. Not only do we write, read and edit code, but code reads, writes and edits us.

Read together, Scott Dexter and Geoff Cox thus offer a sustained argument for coding as a cultural, social and creative practice. In what follows, I will ex-

---

10 | Cox, Geoff: Speaking Code: Coding as Aesthetic and Political Expression, Cambridge MA: MIT Press 2013, p. 8.

11 | Ibid., p. 25.

12 | French philosopher Louis Althusser, in his essay *Ideology and Ideological State Apparatuses: Notes towards an Investigation* describes the way in which ideology functions by transforming individuals into subjects, and it does this via the agents of state apparatuses. Thus, when a police officer hails, or »interpellates«, somebody with a shout of »Hey, you!« and that person acknowledges or »answers« the call by turning round, they are placed in the position of an ideological »subject« and subject to its force. The question arises whether or to what extent, in the digital age, the algorithm is performing a similar ideological function, especially in the realm of social media (see G. Cox: Speaking Code, p. 3).

plore some concrete approaches to coding as performative and dialogic using specific examples of digital concepts as well as digital literary projects.

Live-Coding: performance

A contributor to Geoff Cox's book was live coder Alex McLean. Live coding is a performative writing practice where the coder appears on stage as a live presence and engages in composing lines of code, visible to and readable by an audience, which generate a musical score (see fig. 1).

Figure 1: Live coder Alex McLean on stage

The code is edited in the moment to create new patterns and versions of the score. The performance often involves a dancer who responds to the score that is being generated and leads it in new directions. Thus, the coder-as-live-writer composes and edits their text in dialogue with the machine, in dialogue with the live body of the dancer and in dialogue with the audience. The coder, as with any performer, picks up hints and responses from the audience and adapts the code/text accordingly.

Dialogue within the space of the code

Another site of dialogue within the digital assemblage is located within the code itself. Here, in general, two types of dialogues take place. One we have already alluded to is between developers, where a gloss or commentary is made on the compositional process of the code writing itself. This doesn't impact on what the code can do, but it gives a sense of the developmental process and

underlines Scott Dexter's point about the fundamental sociability and ongoing dialogic nature of code writing.

The second is exemplified in a Nick Montfort/Stephanie Stickland project, *Sea and Spar Between*, a poetry generator based on words or word compounds taken from texts by Emily Dickinson and Herman Melville.[13] Here the source code is used as a space of dialogue between the makers of the piece (Montfort and Strickland) and the user/reader/performer (or whatever we are going to call this figure or subject position). It allows Montfort and Strickland to enter into critical dialogue and give instructions on how to access and navigate this immense work. It also provides contextualising and scholarly material on the source texts. This enables them to engage in an ongoing description of their processual thinking and compositional strategies.

Editing other people's code

A similar process is at work in another of Nick Montfort's projects, *Taroko Gorge*. It is exemplary in a number of ways. Firstly, it is open source code, written by Montfort but available to everybody through creative commons. Having accessed the code, one can simply edit the variables and produce a variant on the original text. The web site for this project[14] displays a number of iterations of this work written by different practitioners, and they are not restricted to English. We can say of this project that each of the iterations is not only in dialogue intertextually with the original but that they are also in dialogue with each other. *Taroko Gorge* then ceases to become the title of a single-authored work and becomes that of a multi-vocal, dialogic, collaborative project.

Permission

A significant digital concept that has a bearing on the notion of dialogues within the machine is that of »permission«. As Charles Baldwin points out in an article in a »Digital Writing« issue of *Performance Research Journal*:

[The command] Chmod sets permissions to read, write and execute directories and files within a directory. To write. To create a file. To edit it. To delete it. A file is written only if permission is given. Web pages are no different. Every file is subject to permission. To read.

---

13 | See the following link to the source code for this project, which also contains the dialogue and editorial comments of the writers: Montfort, Nick/Strickland, Stephanie: http://nickm.com/montfort_strickland/sea_and_spar_between/sea_spar.js
14 | Montfort, Nick: https://nickm.com/taroko_gorge/

To show the contents of a file. To see the name. A file is read only if permission is given. To execute. To execute a file. To run a program. A file is executed only if permission is given.[15]

In other words, it is impossible to carry out even the most basic task on a computer without entering into permissible dialogue with it, without it giving permission for a dialogue and/or an operation to take place. This ties in with Geoff Cox's notion of code as »speech act« and that of the whole digital apparatus as in a state of constant dialogue.

Traumawien

As a quick digression in this discussion about the relationship between writing and coding, coding-as-writing etc., it is worth making a brief mention of Traumawien. Traumawien is a Vienna-based publisher who »considers the paradox of transferring late-breaking digital aesthetics into book form, as new media narrative snapshots of literary genres [are?] otherwise quickly lost in the immense output produced by the web every second.«[16] They are interested in *preserving code within print culture* rather than the output generated by the code itself. Despite its print form, this project thus reaffirms the increasing acceptance of code as a cultural practice with a relevance to the field of literature.

The major point that emerges from this first section of the paper then is that the digital writing which precedes the literary writing in a digital text work has to be understood and accounted for in its own right. A full account of digital literary works will of necessity require analysis of coding as a creative practice. What I have also shown, I hope, is that code is much more than simply a set of executable instructions. It is written in a social, cultural, and political context; it is composed as a set of dialogues; it is a site of dialogue. It is open-ended, susceptible to being constantly edited and revised, and this editing and revision often takes place within a specific mentoring relationship where one developer will act as a »critical friend« to another. Furthermore, under certain circumstances, writing code can be considered not just an adjunct to creative practice, but also a stand-alone creative practice. Coding is essentially performative, dialogic and social in nature. Central to the code writing process is collective and collaborative editing as a way of improving the efficiency of its performance, and it is possible to argue that this mirrors the editing of a literary page-based text as a similar process of rewriting in order to »improve« the performativity of that text, making it more efficacious, i.e. more likely to achieve its intended effect.

---

15 | Baldwin, Charles: »R/W/E or CHMOD – 777«, in: On Writing and Digital Media, Performance Research Journal 18 (2013), pp. 4-9, here p. 5.

16 | http://traumawien.at/about/

3. DIALOGUE AND EDITING IN RELATION
TO CREATIVE COMPOSITION

Having covered the writing processes at work within the machine itself, or rather at the hardware and software ends of the digital apparatus – the compositional and operational modes – we can turn our attention to the interface, to where the digital literary text is actually displayed. I want to begin by flagging up a problem for digital literature with a traditional model of authorship and readership, where these are seen as distinctive functions or figures, before focusing on concrete projects that take on board the potential of digital writing as a site for dialogue and editing in creative practice.

From the early days of theorising about digital literature, the term »reader« has been problematic. It has long failed to adequately capture this function within the digital apparatus. Even when the discourse shifted from the reader as a figure to the reader as a position, it failed to address the radical confusion (or overlap) between writing and reading in the digital literature domain. One solution was proposed by George Landow in his 1997 book on hypertext,[17] in which he coined the term »wreader« in relation to hypertext, as an amalgamation of wr[iter and r]eader. The impulse behind this was to characterise hypertext as a new form of literary production, where the writer and reader both engage in the creation of the text. Nothing new or radical here, of course. The reader has long been viewed as deeply implicated in the creative process, and the idea of the »wreader« is another manifestation of that. However, there are two shortcomings to the adoption of the term »wreader«. Firstly, it firmly locates an engagement with digital text in the literary domain, creating the impression that the »literary« is the logical progenitor of digital writing.[18] This is hardly surprising in that Landow, whose background lies in Victorian studies, is firmly rooted in the literary tradition, and his particular area of interest is hypertext narrative – itself the area of digital writing most closely allied to the literary. Secondly, this term (and others like it) in effect maintain in place the binary opposition of reader/writer and then simply seek to collapse that opposition linguistically. In other words, it looks like an integration of the two terms but no third term emerges from the integration. The words may be agglutinated and overlapping but the functions remain separated. This is contiguity rath-

---

17 | Landow, George: Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology, Baltimore: Johns Hopkins University Press 1997.

18 | In this context I am taking »literary« to refer very roughly to a technology – that of print- and page-based media with its attendant reading strategies, theory of language, canon, etc.

er than integration. We write then read, rather than: in writing we are reading and in reading we are writing.[19]

Writer/readers in dialogue with each other

We have already seen, in the example of *Taroko Gorge* above, an instance of collaborative editing of code and variables to create new but closely related digital texts. Another project that exemplifies the creative potential of collaborative editing, while also touching on the collapse of the distinction between reader and writer discussed above, is called the *Reading Club*, devised and set up by Annie Abrahams and Emmanuel Guez. At each gathering of the Reading Club, four writers are invited to collaborate in the digital editing in real time of an existing text. The »performance« is time-limited to twenty minutes and a video screen capture records the changes/editions that take place over the course of the performance.[20]

In the event in which I participated, the four writers were physically separated from each other. Two were in Paris, one in Colorado, USA and one in Puerto Rico. This sort of digital collaborative writing project raises a number of interesting questions in relation to editing and dialogue.

Firstly, as mentioned above, the roles of writer, reader, and editor are confused in this exercise. The text we were working on was from Raymond Queneau (appropriately enough) and it could be argued that he is the writer and the four participants were simply reader/editors. However, the text that emerged at the end of twenty minutes bore no relation to the original. It was unrecognisable as a Queneau text. For me, an important critical concept which this raised was that of »respect«. To what extent should we, as reader/writer/editors, respect the original by producing a recognisable version of it and to what extent should we edit it beyond recognition? (See fig. 2)

This is particularly pertinent in the case of Queneau, who was an advocate of the endlessly transmutable text. An answer to this may depend very much on the temperament of the participants. To confuse the matter further, we were working on an English translation of the Queneau text, not the original French. Another group of French writers who were working with the original as their starting point, seemed more reluctant to radically rewrite the original.

The question of respect extended to the dynamic interplay between us four participants. There were no ground rules to the collaboration other than the time limit, so the process of editing or rewriting a section which had already been worked on by a fellow participant became slightly fraught. In effect, there was a series of unspoken dialogues between the participants. It was possible

---

19 | There is an equivalent term, »prosumer«, which combines the idea of pro[ducer and con]sumer of a creative work, but that is as ineffective and ugly as »wreader«.

20 | Abrahams, Annie/Guez, Emmanuel: http://readingclub.fr/info/

to determine who was editing the text by the colour of the cursor, and an intriguing element of the exercise was the way in which as a writer you tuned in and out of what the other writers were doing. This felt like a sort of listening. One moment you were sharply focused on editing a particular section of the text and the next you were engrossed in listening to/reading what was going on in other sections of the text. I also found that an agonistic impulse crept in – a desire to intervene and improve upon what others had written and, at the same time, feeling uneasy about having my own variations re-written.

Figure 2: Annie Abrahams and Emmanuel Guez Reading Club
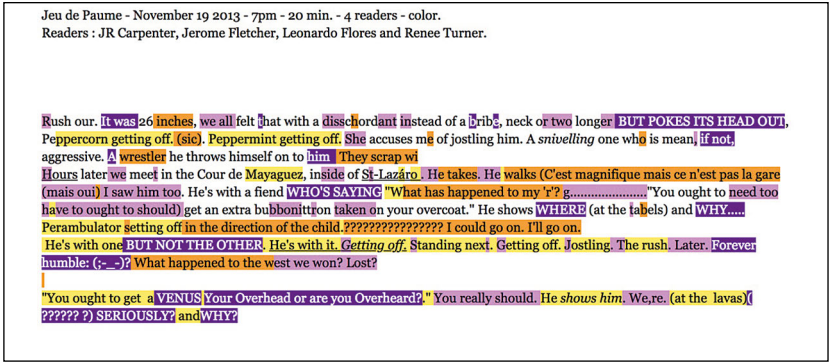
I have been using the term »editing« in the context of the *Reading Club,* but it could be argued that this is not an editing process proper. In the print world, editing has a particular end goal, a telos, where writer and editor are working towards a final fixed text. In this digital writing environment, there is no endpoint. There is no sense of having arrived at a mutually agreed text. It is a time-based writing performance where process is foregrounded. There is no sense of »improvement« as in print-based texts; no sense that the text that emerges after twenty minutes is somehow better than the text at its starting point. Writing, dialogue and editing become an event in themselves, rather than being defined as output.

Texts in dialogue with each other in the digital environment

If the *Reading Club* and *Taroko Gorge* constitute writing projects where writers are in dialogue with/editing code, and where writers are in dialogue with each other through the »editing« of text, there are also examples of digital literature where texts are in dialogue with each other.[21]

---

21 | Another project that involved dialogue between reader-writers that could not be further explored here is *The Reader's Project*, see Cayley, John, Howe, Daniel: http://thereadersproject.org/index.html

This is true of much of J.R. Carpenter's work, which builds upon and expands the affordances of the *Taroko Gorge* text generator. In a work like *Along the Briny Beach*,[22] we see a number of texts scrolling against each other, both horizontally and vertically, and the reader is given a modicum of control by being able to halt or reverse the progress of certain texts, or reveal texts against a contrasting background. The texts themselves are found and well known, including Carroll's *The Walrus and the Carpenter*, sections from Darwin's *Voyage of the Beagle*, Conrad's *Heart of Darkness* and an Elizabeth Bishop poem. In many ways these sort of works can be grouped together under the rubrics of intertextuality, remix, remediation, mash-up etc., all of which are recognised critical categories within literary analysis. The difference is that the texts are in a constantly shifting dialogue with each other, which performatively foregrounds certain aspects and brings to light unexpected links or discontinuities.

A more provocative example of digital texts in dialogue – in this case between digital and print-based texts – is John Cayley and Daniel Howe's work *How it is in the Common Tongues*.[23] Here Cayley and Howe wrote an algorithm that found strings of words on the Internet which when assembled reproduced exactly the Samuel Beckett text, *How it is*. They then printed the text in a hard copy, which scrupulously referenced all the web sites from which they had extracted their fragments of text. They also sent a copy to the Beckett estate, who are notoriously aggressive in protecting the legacy of the great writer. (Beckett famously said of Burroughs' and Guysin's cut-up technique, »That's not writing. It's plumbing.« So I suspect the Cayley and Howe's work would not have pleased him.)

A variation on this project was a digital text, *The Fetch*,[24] created by Kay Lovelace and myself, which was a mixture of original and found texts from which the reader can extract their haunting doubles. (A »fetch« is a figure from Irish folklore, the equivalent of a wraith, a ghostly doppelganger, which appears as a premonition of a person's impending death. The fetch-execute cycle is also the basic operational process of a computer.) As the performer moves the cursor over a line of digital text, it highlights a sequence of words and fetches the same sequence of words plus others pulled from web sites, which may or may not have any connection to the original. Again the dialogue between the two texts performs a series of overlaps and almost surreal discontinuities (See figs. 3 and 4).

---

22 | Carpenter, J.R.: Along the Briny Beach, http://luckysoap.com/alongthebrinybeach/index.html

23 | Cayley, John/Howe, Daniel: How it is in Common Tongues, NLLF Press, Artist's Edition 2012.

24 | Fletcher, Jerome/Lovelace, Kay: The Fetch, http://www.herostrat.us/fetch/

Figure 3 and Figure 4: Kay Lovelace and Jerome Fletcher The Fetch

• 1 • 2 • **3** • 4 • 5 •

ansgressions, to Eros' transubstantiations , to the beloved as doppelganger, as double -dog dare, as flaming conflagration ...

• 1 • 2 • **3** • 4 • 5 •

Arise and walk with me, my daughter. The child steps onto her father's shoes and he takes her hands in his. In this way they establish an equilibrium. She stands on his own two feet, walks in his footsteps, articulated at the sole. A joinery. A journey they begin with slow lumpen steps. As he walks so she walks. A clumsy twosome with the child as doppelganger, as double walker. She backwards as he steps forwards; she forwards as he steps back. Sideways they are more synchronised. More shadow than mirror. Or a reworking of the shadow as mirror. Father and child, in step, present the aspect of Bunraku puppet and master, he hidden in full view. Is this also a reading? This is a reading, a performative reading. Just as the Bunraku chanter holds up the text and bows before it, promising to follow it faithfully. So the shadow of the text is reflected in the mirror and produces a doubling of time. Soon they are formulated by swerve and fluency. A minimal gesture; a clinamen driven by reciprocal supposition.This turn now turns in turn into a re-enactment of the time when the child was gaining a sense of herself as a part of the world around her, when she was finding the extremities of her body, as at that moment when the creature first saw her own toe. And when does the double-walk become a dance? When the pace picks up. When the foot becomes a foot. When the double walker becomes a single-walker. The father lifts their feet higher. The sensation of the effortlessly lifted leg causes her to feel grounded in a no-place. Her desire and gestures are being usurped by her own consent. In this porosity, this fluidity, identity flows between them failing to establish itself on one side or the other. But as the pace quickens they become undone. She slips from his feet and explicates herself in a moment of becoming-alien again.

4. CONCLUSION

What emerges from this short survey is a writing practice that is constituted as an event rather than an object, a process not an outcome. It is a paradigm of an open-ended, multi-vocal performance, which lasts for as long as it is in dialogue with its user/reader etc. and is then dispersed only to be re-assembled and come into being again at each moment of instantiation. What makes digital literature interesting from our point of view is the extent to which it encapsulates a processual and performative mode of writing which brings into sharp relief the notion of the visible and the invisible, the secret and the overt, the concealed and the revealed, the embodied and the ideal. Digital literature is not a thing, not an object. It is an emergent property of a dispersed assemblage of writings within a digital environment. In this respect, processes of dialogue and editing are central. I would argue that where digital writing/digital literature is concerned, rather than having to establish its credentials as dialogic and editable, it would be more pertinent to ask if it is possible for digital writing not to be dialogic and editable.

LITERATURE

Abrahams, Annie/Guez, Emmanuel: http://readingclub.fr/info/

Baldwin, Charles: »R/W/E or CHMOD – 777«, in: On Writing and Digital Media, Performance Research Journal 18 (2013), pp. 4-9.

Carpenter, J.R.: Along the Briny Beach, http://luckysoap.com/alongthebriny-beach/index.html

Cayley, John/Howe, Daniel: How it is in Common Tongues, NLLF Press, Artist's Edition 2012.

Cayley, John/Howe, Daniel: The Reader's Project, http://thereadersproject.org/index.html

Cox, Geoff: Speaking Code: Coding as Aesthetic and Political Expression, Cambridge MA: MIT Press 2013.

Cramer, Florian: »Language«, in: Matthew Fuller (ed.), Software Studies: A Lexicon, Cambridge MA: MIT Press 2008.

Dexter, Scott: »Towards a Poetics of Code«, http://www.academia.edu/2860624/Toward_a_Poetics_of_Code

Fletcher, Jerome/Lovelace, Kay: The Fetch, http://www.herostrat.us/fetch/

Hayles, N. Katherine: »What is e-literature?«, https://eliterature.org/pad/elp.html#preface

Hayles, N. Katherine: My Mother was A Computer: Digital Subjects and Literary Texts, Chicago: Univ. of Chicago Press 2005.

Landow, George: Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology, Baltimore: Johns Hopkins University Press 1997.
Montfort, Nick/Strickland, Stephanie, http://nickm.com/montfort_strickland/sea_and_spar_between/sea_spar.js
Montfort, Nick: https://nickm.com/taroko_gorge/

WEBSITES

http://traumawien.at/about/
https://eliterature.org/pad/elp.html#preface