

## COMBINATORY POETRY AND LITERATURE IN THE INTERNET

FLORIAN CRAMER

### *Permutations*, A WEBSITE

Since I have been asked to present my website *Permutations* at this conference, this paper will first tell what the site is about and then address the issues it might bring up for the discussion of a poetics of digital text.

The website (<http://userpage.fu-berlin.de/~cantsin/index.cgi>) consists of a number of server-side computer programs written in the Perl programming language, each of them reconstructing – and thereby re-inventing – one of a few dozens of combinatory poems written between 330 A.D. and today by, among others, Optatianus Porphyrius, Jean Meschinot, Julius Caesar Scaliger, Georg Philipp Harsdörffer, Quirinus Kuhlmann and Tristan Tzara. Although it is difficult to distinguish a combinatory literature from other forms of literature ever since linguistics defined language as a combinatory system itself, combinatory poetry nevertheless could be formally defined as a literature that openly exposes and addresses its combinatorics by changing and permuting its text according to fixed rules, like in anagrams, proteus poems and cut-ups. Frequently, written combinatory literature does not denote the generated text itself, but only a set of formal instructions with perhaps one sample permutation. Since the poems of Scaliger, Harsdörffer, Kuhlmann and Tzara fall into this category, they turn into something profoundly different as soon as their algorithms are being transcribed from book pages into computer software. The website therefore is an open experiment for finding out what might be lost and gained from such a transcription. *Permutations* is, in my view, not an art project, but rather pataphysics and gay philology.<sup>1</sup>

I Ardua componunt felices carmina Musae  
II dissona conectunt diversis vincula metris  
III scrupea pangentes torquentes pectora vatis  
IV undique confusis constabunt singula verbis.

On the most simple level, the website shows that the history of algorithmic and permutational literature is much older than avant-garde modernism, let alone computer poetry proper. The classical rhetorical figures of chiasm and hyperbaton, the latter also known as “permutatio,” are among the earliest Western prototypes of combinatory poetry.<sup>2</sup> The oldest permutational text adapted in *Permutations* is Optatianus Porphyrius’ *Carmen XXV* from the fourth century A.D.. All words printed in the first and the fourth column of the poem and all words in the second and third make up two sets of words which can be arbitrarily shuffled with each other.<sup>3</sup> The words in the fifth column are fixed, thereby ensuring that the

---

*Date:* October 19, 2000.

<sup>1</sup>In its technical implementation, the website is equally simple. Since all programs run on a server and produce the lowest common denominator of text-only HTML code, it can be read without plugins or additional software in any web browser on any operating system even over slow Internet connections.

<sup>2</sup>[Lie66], vol.2, p.160-2

<sup>3</sup>[Por73], vol.1, p.99

poem will remain hexametric despite its words shuffling. There are 1.62 billion possible permutations of the text. In the computer adaption, the poem randomly permutes each time a button is pressed.

In its initial notation, or state, the poem tells of dysharmonic junctions, uneven meters, rough tones and confused words tormenting the singer. Optatianus Porfyrius, an important formal innovator of European pattern poetry,<sup>4</sup> makes his poem an aesthetic self-reflection which, jumbling its own words, performs and confuses itself simultaneously. Optatianus' *Carmen XXV* became paradigmatic for poetry when Julius Caesar Scaliger coined the term "Proteus verse" for word permutation poems in his 1561 *Poetices*, and made them a canonical poetical form for the century to come.<sup>5</sup>

Scaliger's example line, "Perfide sperasti divos te fallere Proteu" ("Wickedly you hoped to deceive the gods, Proteus"), was the prototype of countless poems in the 17th century whose lines, written either in Latin or in one of the new national languages, contained words to be shuffled. Not unlike Optatianus' *Carmen XXV* whose permutability was restrained through the fixed words in its fifth column, Scaliger's line can not be jumbled at will if the hexameter is paid attention to. The poetical permutation of the six words therefore doesn't map the mathematical permutation of six ( $6! = 720$ ). The difference between poetical and mathematical laws of permutation was abolished in the 17th century when the perception of Scaliger coincided with a renewed interest in the "ars" of Raimundus Lullus and the Christian Kabbalah. While Lullus used combinatorics to generate ontological and theological statements, 17th century science rewrote Lullism into a generative systematics of encyclopedic knowledge. Thomas Lansius, Georg Philipp Harsdörffer, Quirinus Kuhlmann were at once scholars, language researchers and writers of Proteus poetry.

Aside from two Proteus poems (both of which are adapted in *Permutations*), the 17th century poet Georg Philipp Harsdörffer wrote a morphological word generation machine he called "Fünffacher Denckring der teutschen Sprache" ("Fivefold Thought Ring of the German Language").<sup>6</sup> Each of its five concentric circles contained a set of morphemes which, in their combination, were supposed to cover all existing and potential words of the German language. Harsdörffer's "Denckring" not only expands on Lullus, but also on the 17th century linguist Justus Georg Schottelius who considered the combinatorics of one-syllable "base words" ("Stammwörter") the principle of the German language. Schottelius anti-nominalistically conceived of them as words which "mean their thing right away" and believed them to be derived immediately from the Hebrew and divine language.

While the Proteus poetry of the 17th century employed combinatorics as a means of calculation and control, the artistic avant-gardes of the 20th century reinvented same or similar poetic forms as part of a poetics of indeterminacy and chance. Tristan Tzara proposed to create Dada poetry by cutting out the words of a newspaper article, shuffling them in a bag and writing them down in the accidental order they had been pulled out.<sup>7</sup> Despite the anti-art gesture, Tzara's instruction to select, break up and permute a group of words exactly conforms to Julius Caesar Scaliger's definition of the Proteus verse. Between Scaliger and Tzara, there however is not only a shift from determination to chance, but also from closure to openness of the system. All pre-20th century permutation poems shuffle a fixed set of data directly inscribed – hard-coded – into themselves, but Tzara's Dada poem merely denotes a process which can be fed with arbitrary data. By allowing to

<sup>4</sup>A comprehensive history of pattern poetry is given in [AE87]

<sup>5</sup>[Sca61], no pagination

<sup>6</sup>[Har36], vol.2, p.517

<sup>7</sup>[Tza75]

take any Web page as the input data, the computer adaption of the poem even radicalizes this difference; the process now involves a bigger repository of text, happens in real time and, by algorithmic automation, doesn't require any manual work or skills on behalf of the reader.

*Permutations* finally include some self-invented automata, such as "Here Comes Everybody", a processor of James Joyce's "Finnegans Wake" which algorithmically mimics the portmanteau word poetics of the novel. Hyphenating its text and recombining the syllables according to stochastic probability, the program perpetually creates new texts with newly generated portmanteau words from the novel. John Cage's radio play "Roaratorio. An Irish Circus on Finnegans Wake" formally processes the novel in order not to expand, but to reduce the volume of text.<sup>8</sup>

#### LANGUAGE COMBINATORICS AND COMPUTER TEXT

Without doubt, it is philologically incorrect or problematic at least to rewrite pre-digital combinatory poetry into computer programs. The transcription potentially blurs the difference between an anti-nominalist, theologically and hermetically influenced linguistic thinking of the Renaissance on the one hand and the concept of language as arbitrary material in avant-garde modernism on the other. Juxtaposing both discourses, the website however shows that any contemporary perception of the Renaissance texts is inevitably triggered and filtered through the knowledge of avant-garde literature, computer poetry and literary theory. If both traditions therefore influence each other, the opposite conclusion must be drawn as well: Any concept of digital literature which does not reflect language combinatorics and algorithmically processed language is severely restrained.

On a purely formal level, the combinatory poetry of both the Renaissance and the 20th century has a common set of features which as well seem to be relevant for a poetics of literature in computer networks:

(1) Densification

An compact source code (instruction set) generates an abundance of text.

(2) Micro-grammar

Reproducing the linguistic mechanisms of word and sentence creation, combinatory poetry is a generative reflection of language.

(3) Filtering

Combinatory poetry uses formal methods to process language and transform text. It thereby shows that the poetic potential of computing machines is not limited to transmitting ready-made signs. Computers are not merely a transport devices, but potential senders and receivers, writers and readers of text as well.

Since a computer can act at any point of the communicative process, it is not simply a medium – i.e. an instance *between* a sender and a receiver –, but a universal semiotic machine. Misreading the computer as a mere medium, humanities have wrongly assumed that their studies of the computer have to be "media studies" (instead of semiotics). Likewise, computer art was misunderstood as so-called "media art." A result of this misreading is, as it seems, that concepts and methods developed by media studies since Kracauer and

---

<sup>8</sup>The method to expand text through stochastic algorithms has been frequently used since the 1950s when Theo Lutz and Max Bense produced computer-generated variations of Kafka's prose (as described by Reinhard Döhl in [Döh98]). Markov chains have been prominently used in poetry by the literary scholar Hugh Kenner and the British poet Charles O. Hartman. They are also used in MS/DOS program *POE* by the Austrian poet Franz Josef Czernin ([Cze97]) and in Ray Kurzweil's *Cybernetic Poet*.

McLuhan for analyzing film, television, radio and video were plainly reapplied to computers and the Internet. As a consequence, notions like “multimedia,” “interactivity” and “nonlinearity” have been mapped from TV and video onto digital literature. While it is of course useful to distinguish a movie as linear form (of a reel whose time and sequence of display can be exactly determined) from a computer game as a nonlinear form, the same distinction fails to describe a literary text whose perception might be rather linear or rather not depending on the way an individual reads it. While “new media” notions derived from film, TV and video made little to no sense in literary theory and studies of digital code, the conceptual confusion they left still persists and continues to obstruct critical debates.

From the viewpoint of a computer programmer, the text generators that make up *Permutations* may be primitive. But making their algorithms transparent, they make readers pay attention to the fact that any digital text – and any digital poetry – is potentially machine-executable, a sequence of signifiers which, beyond merely relying on computer systems, actually sets them up. I thus consider the website a modest statement against equating network computers with simple transmission media and typographical interfaces, against mistaking the web browser for the net and against restraining computer network literature to so-called “hypertext” and so-called “multimedia.” While it might seem that, in comparison to the latter, generative text has remained a marginal form of digital literature, a more thorough consideration should take into account, for example, machine-generated invoices, automated bank statements and official letters, Internet search engines, “personalized” portals and home-order catalogues, not to speak of fully automated control and regulation systems in industry production, aviation and on the stock market. They all exemplify how efficaciously algorithmically manipulated writing has intervened into everyday language and culture; a status quo which the concepts of “hypertext” and “multimedia” don’t reflect at all. Instead, computer viruses like *Melissa* and *I LOVE YOU*, small bits of text written in computer control code, strike me as perhaps the most dense and interesting examples of contemporary literature in the Internet. Viruses at once follow and extend the combinatory design principle to create an abundance out of few signifiers by infection, self-replication and mutation of code. They could make other writers in the Internet aware that the mere syntax of the code they use is of explosive virulence, all the more when global technical infrastructures depend on it.

This should make it clear why “hypertext” is anything but an exhaustive or general concept of digital textuality. Nevertheless, “hypertext” used to be both the coded format and the aesthetic program of much if not most literature in the Internet.<sup>9</sup> While it would be aesthetically naive of course to expect all digital literature to be written in program code, it seems reasonable to expect from net literature that it conceptually and aesthetically reflects the semiotic and technological conditions of the system in which its signifiers flow. Until recently, this expectation was rather met by poets who didn’t call themselves poets, but “Net.artists”, rooting themselves in conceptual art rather than in literature.<sup>10</sup>

Recently, the “codeworks” poetry of mez (Mary Ann Breeze), Alan Sondheim, Ted Warnell and others has taken up impulses from Net.art by incorporating ready-made bits and syntax from programming languages, binary machine code, network protocols and

<sup>9</sup>The implication of “hypertext” as a hypertrophy of “text” is not only questionable, it all the more contradicts the fact that the “hypertextual” World Wide Web just forms the utmost and least general code level of the Internet.

<sup>10</sup>Such as *jodi.org*, *I/O/D*, *Mongrel*, Heath Bunting, the *ASCII Art Ensemble* and *0100101110101101.org*. Comprehensive material about Net.art is available in [Bau99] and [WD00].

markup conventions of interpersonal network communication.<sup>11</sup> Contrary to expectations that net literature would increasingly become multimedia, these codeworks circulate as plain E-Mail. Not being algorithmic in a strict sense, they nevertheless play with the fact that they might be read as (potentially harmful) machine code, and achieve densification, micro-grammar and filtering by hybridizing human and machine languages. If codeworks could thus be called a post-combinatory poetry, I hope the gay philology of *Permutations* provides material against which the “post” prefix may be matched.

## REFERENCES

- [AE87] Jeremy Adler and Ulrich Ernst. *Text als Figur. Visuelle Poesie von der Antike bis zur Moderne*. VCH, Weinheim, 3 edition, 1990 (1987). 2
- [Bau99] Tilman Baumgärtel. *net.art*. Verlag für moderne Kunst, Nürnberg, 1999. 4
- [Cze97] Franz Josef Czernin. (Vortrag über das Programm POE). Unveröffentlichtes Manuskript, 1997(?). 3
- [Döh98] Reinhard Döhl. Von der ZUSE Z 22 zum WWW. Helmut Kreuzer zum 70sten, 1998. [http://www.stuttgart.de/stadtbuecherei/zuse/zuse\\_www.htm](http://www.stuttgart.de/stadtbuecherei/zuse/zuse_www.htm). 3
- [Har36] Georg Philipp Harsdörffer. *Mathematische und philosophische Erquickstunden*. Texte der frühen Neuzeit. Keip, Frankfurt (Nürnberg), 3 edition, 1990 (1636). 2
- [Lie66] Alfred Liede. *Dichtung als Spiel*. De Gruyter, Berlin und New York, 1992 (1963/66). 1
- [Por73] Publilius Optatianus Porfyrius. *Publilii Optatiani Porfyrii Carmina*. ?, Turin, 1973. 1
- [Sca61] Julius Caesar Scaliger. *Poetices libri septem*. ?, Lyon, 1561. 2
- [Tza75] Tristan Tzara. Pour fair une poème dadaïste. In *Oeuvres complètes*. Gallimard, Paris, 1975. 2
- [WD00] Peter Weibel and Timothy Druckrey, editors. *net condition*. MIT Press, Cambridge, Mass., 2000. 4

---

<sup>11</sup>The term “codeworks” was coined by Alan Sondheim. The September 2001 issue of the American Book Review will feature a number of critical essays on codeworks.