

THE SOFTWARE

LANGUAGE

AR

T



EVEN though authors of new media theories do not pay much attention to the digital textuality and digital literatures (they find more new media specificity in web sites, digital cinema, computer games, and in virtual reality installations) we can consider this domain to be explicitly a new media one. First of all we mean the digital literatures after the hypertext—as well as simultaneously to it—that can be called second order digital literatures, software poetry and (software) language art. These designations refer to the expanded concept of digital textuality meant for artistic spaces and artistic interventions; this textual practice is by no means a continuation of the literature-as-we-know-it, it is a textual praxis, close to the new media specificity. Among its readers-users may take place those who are keen on neither Dostojevski nor Oscar Wilde nor Charles Baudelaire, but are computer game geeks and participants in the DJ and VJ culture. In this paper, we focus on the expanded field of digital poetry, which in technical terms can be called the software language art, and which we believe belongs primarily to the field of software art as one of the most characteristic forms of new media art. We will therefore first describe the main cultural turns and the paradigm changes that come with the new media art.

### **From Work of Art to Service of Art**

The shift of the emphasis from the (industrial) production and manufacturing artefacts to the services sector in the economy of post-industrial societies also affects the contemporary art; its

social position benefits also from other shifts in contemporary (postindustrial, information, software) societies — an example would be the increasing stress placed on knowledge, innovation, education, use of new technologies, communications and spectacle. We are entering a world, in which data and intangible, abstract entities, immaterial products and services, mobility, flexibility, decentralization, rhizom-like order of organization, and high-level professionalism are gaining importance. The role of the national state is at stake in the globalized networking based politics, more and more affected by the multinational capital and international institutions, and similar shifts can be observed in science (destabilization and relativization of the concept of subject independent nature, i.e. objective nature, natural laws and objective truth) and in new economy (the shift from the tangible wealth towards services and information).

By taking a look at the different movements of the contemporary art, especially the new media art, we can find out that material, stable artefact is under similar pressure as the national state is in the contemporary politics, material, tangible wealth and production of artefacts in the new economy and the so-called objective nature in techno-sciences. The art is also less and less about producing completed products that are “kunstwerks” by nature, and more and more about processes, immaterial entities, relations, performaces, software and services. This shift does not occur only with the contemporary art projects that are no longer works of art in the traditional sense, but “would-be-works-of art,”<sup>1</sup> it is also documented in theory (“the end of art” issue as it was discussed in Heidegger’s, Benjamin’s, Danto’s, Groy’s and Kuspit’s theories) and in the poetics of contemporary artists. We then encounter art that increasingly exceeds the manufacturing of artefacts and is crossing over to a domain that can be called a service of art — meaning a part of contemporary art (especially the new media one) is crossing into the service sector of (new) economy in the postindustrial, information, spectacle and software societies. Its services are equal to those in the domain of education, management, counselling, finances, politics

etc.; they are then equal to the activities based on knowledge and professionalism and are as flexible as possible.

Contemporary, especially new media art as a service of art directs us to the question “what is a service?” It is by no means an artefact, a completed product, it is essentially an activity, a praxis, a process, an exploration, an intervention (inside things, states or processes). The service is not so much the manufacturing of things as it is a process of reshaping the thing, moving it, connecting it and incorporating it into new relations, (re)combinations and (re)contextualizations. The service presupposes a problem, a challenge or an order to be solved or executed. The performer of the service is always faced with a certain task, challenged to solve it in a sequence of steps, chosen as economically as possible. The service therefore ends with a solution of the problem (or its removal) and not the manufacturing of an object.

The service, understood in the sense of professionalism demanded by the information society, always presupposes a procedure that has to be as rational as possible, economical, divided into phases, steps, operation commands needed for it to be carried through. This kind of procedure—an exactly defined, planned procedure, executed through an economical sequence of steps—is called an algorithm. The algorithm has for quite some time no longer been an exclusive domain of the mathematical operations, it is the core of all sophisticatedly defined processes intended for performing certain tasks, solving problems, researching the state of things etc. It would not be an exaggeration to say that artistic services are algorithmic by nature; by the moment art begins to position itself beyond the aesthetic and becomes oriented towards tasks, research and problem solving, it is forced to carefully elaborate the procedures and to define the instructions to be carried out in order to get to the solution quickly and through economical phases. Those artistic services based on the use of computer technologies, intended for algorithmic functions, are especially and explicitly algorithmic.

A lot of new media art projects can thus be understood as interventions and services within different states of things, they have an

algorithmic nature and are often stimulated by non-artistic motives—for example with regard to political, research and communication needs and interests. The work done by such an intervention is adequately articulated solely in its documenting—as Boris Groys has pointed out in his text *Art in the Age of Biopolitics / From Artwork to Art Documentation*. After having mentioned different forms of artistic interventions in the everyday life through the attempts to form unusual life circumstances, he wrote: “None of these artistic activities can be presented except by means of art documentation, since from the very beginning these activities do not serve to produce an artwork in which art as such could manifest itself.”<sup>2</sup>

The service, connected to the executing of the task, constitutes the artist as the performer or executer of the service, but at the same time often includes also the person who had placed the order for the service or at least the person who had initiated it. An example of this are the contemporary curators and art directors of big festivals and exhibitions, who—along with the “Call for Entries”—often also define a theme to which the artists are supposed to respond with their practices. As an example of such an order we can mention the CODeDOC project (2002) of the Whitney museum in USA (the project went on in the *Ars electronica* festival in Linz next year): the curator Chistine Paul issued a call for software tenders, dependent upon an exactly defined order. The participating artists were prescribed the choice of programming and scripting languages, the code had to move and had to connect three points in space, could not exceed 8 KB and had to be interpretative. The transfer from the artefact to the service of art and the artist as the one who executes the service (the service depends on certain instructions, software, algorithmic approach) is also on its way to abandon the metaphysics of artistic creativity and genius. The artist as the one who executes service, performs certain tasks, solves problems, does research, defines commands, executes algorithms and does not wait for the divine inspiration to come upon her.

The artistic service actually moves art closer to the new economy, that has customization as adaptation of the service to the user's preferences as one of its key concepts. The power to control, to navigate, to form and to finalize that in the traditional paradigm belonged exclusively to the author, is now being transferred also to the user; the term "user friendly", although worn out and trivialized, does have a certain content. It is by no means solely the artefact that is customized—it can apply to the service as well: a software artist can, for example, create a program as an open—as much as possible—scheme to be concretized in finalized by the users, according to their personal preferences.

### **Texts intended for experiencing the cyberverb**

In this paper, we will focus on text projects, generated with the state-of-the-art software, that can be treated as a part of software art. Nowadays the latter is the main field of Internet art—it is its second phase if we consider the following classification by Alexander Galloway: "Early Internet art—the highly conceptual phase known as "net.art"—is concerned primarily with the network, while later Internet art—what can be called the corporate or commercial phase—has been concerned primarily with software."<sup>3</sup> This differentiation is close to the one Lev Manovich makes when he distinguishes the Internet art of the 1990s from the software art of the beginning of the 21st century, typically defined by a generation of artists who are also active in the field of Flash programming. "This generation is no longer interested in "media critique" which preoccupied media artists of the last two decades; instead it is engaged in software critique. This generation writes its own software code to create their own cultural systems, instead of using samples of commercial media."<sup>4</sup>

Software art is explicitly a service profiled art, using algorithmic procedures, intended for solving problems. The question that needs to be asked is what happens to the text in the moment we enter the world of services and agony of completed stable artefacts, and tangible works. By no means, it is no longer the great text such as the

“textwerk”, it is a fragment, a patch, a (short) program or its modification, textual sequence as a moving target in a computer game (e.g. *The Trigger Happy*), a digital poem, a poetic generator or simply a software controlled scheme, meant for the user’s concretizations or customizations. The question of genre and form is becoming more and more secondary; we encounter hybrid and temporary forms, texts as textscapes. As a rule, their function is no longer storytelling, instead they serve as a demonstration of a new way of experiencing the verbal in the world of on-line communication and inside the software paradigm. It seems a series of software and digital poetry (especially animated, kinetic) pieces answer the question of the role of a word and text in the information and software societies. Why (still) a word and not simply attractive images and sounds? The software language art as a service is therefore meant for a new definition and demonstration of cyberword and cybertext (this may also happen in a form enabling customization).

Projects (and textual services) of software language art are nowadays often Flash generated; it seems a parallel, similar to the one Galloway and Manovich establish in Internet art can be found in the field of digital textuality. In Internet art we have first the net art with the reference to the networking, followed by software art, whereas in the digital texts there is a gap between the hypertext with a reference to the links and the software language art, which is often the work of authors of the Flash generation. What is characteristic for the Flash generated software language art that enables a truly special textual visuality and aesthetics?

The issue here are by all means no longer hypertextual texts with links and lexias, and with the suspense accompanying the clicking of the underlined words (“words that yield”) as hyperlinks. Neither is at issue the enjoyment of uncertainty and the feeling of being lost in a labyrinth that accompanies the works of hyperfiction (for example works of Michael Joyce and Shelley Jackson) which are often intentionally designed as actual labyrinths demanding a sophisticated search for passages and really risky solutions. In Flash gener-

ated language art most of the action is focused on the word-image-body-movement as the basic unit of such textuality, which is as a rule kinetic—meaning that it foregrounds a signifier, as visualized and as animated as possible, in the role of a software controlled “parola in liberta.”

One of the crucial demands accompanying this sort of texts is connected to the nature of the film way of showing or demonstrating things in the present, to what Lev Manovich has expressed as “a general trend in modern society toward presenting more and more information in the form of time-based audiovisual moving image sequences”<sup>5</sup>. It is important that the text-film (in the Flash, modified by the author-programmer) is short, not unlike a music video its duration is for example 2’45”, but it is defined by an extreme density of information, action, it is a world, compressed into the attractive textual “music video”. As examples of such texts we may mention the Flash generated poem of Claire Dinsmore *The Dazzle as a Question*<sup>6</sup> and Brian Kim Stefan’s animated poem *The Dreamlife of Letters*.<sup>7</sup>

In the traditional as well as in the modern and contemporary print-based poetry, the definition and criteria judging over literary forms are connected with those aspects of the texts concerning the content, motif, syntax and organization. We come across lyric and epic poetry, free verse and verse with rhymes and assonances, there are also forms such as the sonnet, triolet etc. However, in digital (software) poetry the instance that generates the forms and judges them, is connected with the programs used. Digital poetry is actually software poetry, that is to say it is poetry generated by very special programming and scripting languages and their modifications (“poetry patches”), for example Perl, Java script, Flash, Shockwave, Director etc. In the present some poets themselves define their poems as poems of the Flash or Shockwave Poetry (for example Komninos Zervos and Giselle Beiguelman); there are also online courses in Flash Poetry which means that this sort of textual practice is already established as a genre of its own. And in a moment we are beginning to talk about specific software based



poetry all the software concepts and devices are to be considered also as literary devices; rather than issues of style and metaphor the functioning of mouse(over) event is being crucial in Flash poetry. When we are mentioning the digital poetry that presupposes the destabilization of verse by applications of non-linearly distributed verbal and non-verbal components, and frequent reduction of poetic language just to nouns, one needs to emphasize that the “poetic” is now beyond the lyric as understood by the movements of modern and contemporary poetry. In digital poetry, too, we can sometimes still discover the making of pure “poetic atmospheres”, the tension between the said and the unsaid, the written and the whiteness (in kinetic and animated poetry, for example), now revealed through the loops between the text that is already in our field of vision, that is, displayed, and the text that is yet to appear, however one of the striking features of this poetry is first of all its inventive work in the field of broadening of concept of poetic language (or even language at all). With the latter we refer to a language suited to post-lyrical sensations and attitudes of the post-lyrical subject and the subjectivity of the “mix, cuts & scratches mind”. If we want to define a reference framework for the postlyrical kinetic digital poetry, and even so called software language art on the level of contemporary popular culture and its audience then this is mainly the club, DJ, and VJ culture, the on-line Internet culture and the verbal, netspeak based culture emerging from the on-line and mobile communications. Rather than reading Baudelaire’s, Whitman’s and Rilke’s lyrics the authentic audience of kinetic digital poetry is familiar with net art, software art and electronic installation art as well as the genre of moving images.

Flash language art also enables the reader, in the role of the user, to have a very creative, intensive, even an intimate contact with the text—Deena Larsen, in her digital text *Carving in possibilities*<sup>8</sup> expressed this characteristic with the demand to “sculpt again” and not “read again”. Devices such as scroll-bar, mouse and stylus (when using a palm pilot) enable the reader to handle the written

in a very specific, intimate way and to interfere with the text through an interface, such as a screen covering the text like a curtain, which responds to the click of the mouse (within mouseover event). In Larsen's *Carving in possibilities* words are hidden behind the surface—like the objects, provisory wrapped by the artist Christo in his land art projects. The reader is asked to find—by the means of her “mouse event” procedure—the covered/”wrapped” words and make them appear on the screen. By “mouseovering” an image of shapeless stone is being transformed into Michelangelo's David. User's action is individualized, the sequence of textual components adapts to her interventions (this is “customization” as a procedure known from the new economy) and it always produces or sculpts a different succession of the written, that is to say, accomplishes a different textual event. Larsen's opening line “I saw precisely what the stone was meant to be” is a starting point for various textual continuations/derivatives caused by random repositioning of the mouse-touch on the screen.

We say event (also in Flash vector-based art is talked about mouse event), and digital poetry really is about the event, it is about making the text with a stressed temporal feature, based on two levels—on the internal “unwrapping” of the textual hidden layers as well as on the reader's/user's reading in the form of interactive intervention into the texts (which is often the case). Text as an event implicates a textual life, which is a form of an artificial life (also in the meaning of replicating certain textual components in the textual postproduction, reproduction and interactive reading). It is essential that the components of a text are not based solely on words but above all on relations among words and on special atmospheres connected with these relations. The author-programmer of digital poetry is therefore the one who is able to insert word materials into very special relations that are highly shifted from the known relations (from the everyday language or the profane marketing and advertising verbal communication). Her role is not merely the saying of the “poetic words”, it is above all arranging the stage of rela-

tions among words and even within one single word. Therefore the digital poetry text (designed as an object, browser, textual ambient, project, piece of software...) appears to be an eminent linguistic work of art, to which the demands of the new media aesthetics and poetics — such as digitality, software, logic of database, mosaic nature, networking, customization, aesthetics of flatness and nearness, sense of the game mode, kinetics and multi media — are crucial.

A part of software language art are also texts based on the code language, or perhaps those are hybrid products in which the letters from natural languages are mixed with characters from scripting and programming languages. Such example are the texts of the Australian author Mez and partly also the works of Alain Sondheim and Talan Memmoth. It is obvious that their “written form” can be very provocative as well; it is not just the computer execution of the text that is worth our attention but also the text itself.

This kind of code texts are a big challenge also for the readers/users who are forced to develop a sophisticated “mental interface”, needed to decode such texts not suitable for the quick linear reading — on the contrary, they require an effort, a sense for associations, paying attention to the divided/broken words and their units, recombining fragments etc. In the scheme author — text — reader the emphasis is now undoubtedly moving toward the reader as a user, stimulated by such texts to take an attitude — not unlike the DJs and VJs in their production procedures. It seems authors of software language art are with their texts indeed simply providing material and tracing schemes; a lot of work is left up to their readers — in a certain way this challenges and stimulates the present “mix, cuts & scratches mind”. It is therefore not only the author who faces the task demanding the ability of algorithmic thinking, the same task awaits the reader who can also be rejected when she encounters such texts — like in a computer game. To be successful, she has to perceive the text in a very complex manner, to decode it she has to generate an actual algorithm, she often has to read even with a pencil or a stylus in hand, to sketch and write down the steps she has already taken during her encounter with the text.

It is also of importance that the software nature of such texts—i.e. the code language, by no means a mere instrument on the way to the meaning or machine-based execution, but with its own value—is also emphasised. What do we mean by that? “In poetry, says Jacobson, words are not simply strung together for the sake of the thoughts they convey, as in ordinary speech, but with extraordinary attention to patterns of similarity, opposition and parallelism created by their sound, rhythm and semantic connotations. Literary language, that of the poetic text, proclaims its material being, inviting attention to itself as an acoustic and graphic substance, rather than functioning as an invisible glass passage to meaning.”<sup>9</sup> When talking about programmable medium, attention similar to that paid to the “patterns of similarity, opposition and parallelism” should be paid to the software used—meaning the reader is faced with the task of reading through a double optic; she reads the text as a text generated by executing certain commands, but at the same times she turns to its code. A successful reader of texts belonging to the software language art is above all the reader who sees (recognises, deciphers) a lot also in the field of the code; in a certain sense a reader-programmer reads over more than the one who does not possess such skills

.

- <sup>1</sup> Strehovec, J. "The Atmospheres of Extraordinary in the Installation Art", *A-r-c*, Issue 3, (November 2000), [http://a-r-c.gold.ac.uk/a-r-c\\_Three/texts/3\\_janezo5.html](http://a-r-c.gold.ac.uk/a-r-c_Three/texts/3_janezo5.html).
- <sup>2</sup> Groys, B. "Art in the Age of Biopolitics. From Artwork to Art Documentation". *Documenta 11\_Platform 5: Exhibition. Catalogue. Ostfildern -Ruit: Hatje Cantz Publishers, 2002, p.108*
- <sup>3</sup> Galloway, A. "Protocol" -Excerpt from Chapter 7 "Internet Art". *Rhizome.org*. <http://www.rhizome.org/thread.rhiz?thread=12700&text=24436#24436> , 2004
- <sup>4</sup> Manovich, L. "Generation\_Flash". <http://www.manovich.net/>, 2002
- <sup>5</sup> Manovich, L. *The Language of New Media*. Cambridge, Mass.: The MIT Press, 2001, p.78
- <sup>6</sup> Dinsmore, C. *The Dazzle as a Question*. <http://www.studiocleo.com/projects/dazzle/index.html>
- <sup>7</sup> Stefans, B.K. *The Dreamlife of Letters*. <http://www.ubu.com/contemp/stefans/dream/index.html>.
- <sup>8</sup> Larsen, D. *Carving in Possibilities*. <http://trace.ntu.ac.uk/frame6/>, 2001
- <sup>9</sup> Wyman, S. "The poem in the painting: Roman Jacobson and the pictorial language of Paul Klee", *Word & Image*, Vol. 20, No.2 (April-June 2004), p. 139