

Object Oriented Poetry

Index

Object Oriented poetry

Overview about Object Oriented Programming

Questions of Structure

Artificial Memories

The right Formulas to compare

Bibliography

Object Oriented Poetry

The purpose of this essay is to make a literary study on the object oriented programming paradigm, defining an Object Oriented Poetry. I will compare two kinds of text: one is the executable text and the other is the literary. This is to show how literary and rhetorical categories are actually used for writing object oriented code and why. The discussion is possible because high -level programming languages, like Smalltalk, Ruby and Python, are made to be more human readable in order to favour a fruitful communicative exchange between the machine and the person who interact with it. I will study object oriented paradigm at the level of language form, and I will give an overview of it's execution by an interpreter, in order to connect it with the results of the execution of the program.

Overview about Object Oriented Paradigm

The first kind of text I am going to compare belongs to the Object Oriented Programming Paradigm. The fundamentals concepts of it can be used in all the mentioned programming languages to develop software code. One of these concepts is "the object", which is defined by others, like the concept of "class", and "method". At first I would like to reflect on the concept of *class*: it defines the characteristics of an "object" or an "instance";

"Classes provide modularity and structure in an Object-oriented computer program. A class should typically be recognizable to a non-programmer familiar with the problem domain, meaning that the characteristics of the class should make sense in context."

The definition employs a process of contextualization that lets us understand why this programming language is human readable. This *context* is effectively a metalinguistic category. In describing this object oriented programming paradigm it is easy to have the impression that it could be a logic process described through a strict syntax. For example classes can be of different specialized types and work in parallel with other classes, in this case they are called subclasses and the principle that defines them is called inheritance. Classes and subclasses contain text code and are used to define object behaviour with encapsulation.

The objects are giving the name to object oriented programming paradigm because they are the most innovative concept of it. They are much powerful and really generic data structure. They take their behaviour from interfacing with other data structures. A data structure, in the case of objects, is defined by the class the object belongs to, and by one or more methods included in the class. The possibility to use methods in regards to a large range of object types is the main characteristic of object oriented programming languages. This is called Polymorphism. It provides a generality in writing object oriented code that enables these "objects" to be used in many different digital context and systems.

This, for example, is a characteristic of open source software, which has to face the existence of many different release of the same operating system. And the use of open source software for managing network traffic is the basement of the construction of web based application. But however what is important to underline in this moment is the connection between object oriented programming language and the logical process of abstraction.

A formal trace of this connection, for instance, could be observed in the Boolean operators (is, and, or, not), in the class or subclass, which define a data structure. They are made to be as simple as possible, but they are still corresponding to an abstract scheme of possibilities, the one defined more generally by Logic or Mathematics. Their form does not really care about the fact that logic is a theory of human interpreting abilities. They act in the context of a computation of memory that can effectively realize what the text is supposed to represent. Therefore it is difficult to make an attempt of defining a theory behind object oriented programming languages; because the formal definitions that this programming paradigm is built by are strict in syntax, but very general and obscure in the meaning.

In conclusion the data structures we are talking about are able to deal with images, video, and every kind of multimedia digital file, by simply abstracting it into syntax values. This is what Bill Viola says in an article about video and data structures:

"Soon, the way we approach making films and videotapes will drastically change. The notion of a master edit and original footage will disappear. Editing will become the writing of a program that will tell the computer how to arrange the information on the disc, playing it back in the specified sequence in real time or allowing the viewer to intervene. [...]

We are proceeding from the models of the eye and ear to models of thought processes and conceptual structures in the brain.[...]

Today, there are visual diagrams of data structures already being used to describe the patterns of information on the computer videodisc.¹

I would like to take under consideration the article of Viola for two reason; the first one is that it focuses on the ability of programming languages to manipulate video/images. This feature of digital systems has the characteristic to involve one of the main arguments of the theory of cinema profiled by Deleuze in *Cinema 2, the time-image*. An image, he says, can contain a linguistic utterance, but the message it sends is larger and regards specific particulars of the narration in its context. The discussion of Deleuze starts from the image as representation and explains how the linguistic syntax is not able to represent all the particulars of the image that concur building meanings in it.

¹ Wardrip-Fruin N., Montfort N., *The new media reader*, Cambridge Mit press 2003, B. Viola 31. *Condominium in data space?*.

The point of view of the object oriented programming paradigm, in this case, is the opposite one: it starts from language, and focus on its possibility to generalize enough to include an image, as a general object in its syntactic language system. A computational memory represents images as a sort of value, or syntactic unity. It totally misses the level of interpretation and subjectivity that Deleuze indicates as essential to understand the meaning of the image in the language of cinema. The point of view of Deleuze is, indeed, caused by the fact that he makes a semantic analysis of the images in cinema. Starting from the point of view of object oriented programming languages will implicate instead to consider language and sign from a syntactic point of view, and it has to own a causal connection with a representation, or an input.

Questions of structure

It is convenient to point to some theoretical references to explain the methodology I will use in my comparison. In the fields of literary studies speaking about the structure of a literary text is not so simple. There are much more language variations than anywhere else. In general we could say that the structure of a literary text is defined more by its meaning than by its form, if not for some ancient exceptions. Rather on the contrary the meaning of a software text is strictly related to its form. If the output of a software can be considered the meaning of it, it is only because a human mind can interpret it, and not because it is interpreted by the digital system itself.

The most interesting research in semiotics for developing the idea of object oriented programming paradigm and poetry, is Peirce's classification of signs. It not only involves abstraction, but it even speaks about a clear connection between interpretation and action, which is useful in the attempt to describe the dynamic of meaning for the encoded languages.

"Sign: anything which determines something else (its interpretant) to refer to an object to which itself refers (its object) in the same way, the interpretant becoming in turn a sign, and so on ad infinitum."²

The *interpretant* used by Peirce to define the sign is not the same this as the interpreter of python programming language. The interpreter is to be understood as a sign of the programming language itself, which causes an execution. The *interpretant*, as defining the meaning of the sign and in the contest of a digital memory, can be attributed only to the execution of an operation, its output. In the description of the nature of sign Peirce points to a casual relation between the idea represented in it and the thing the sign is referred to.

As J. Kristeva points out, Peirce's Theory of Sign does not include a subjective perspective outside the language system. She sustains that this is a literary limit for the semiotic analysis. I am not convinced that her argument is really constructive, in a literary context, but I can see the validity of it in a general perspective on sign. Bakhtin's papers *The discourse in the novel*, for example, seems to me to state differently the meaning of the word from the interpretation that Kristeva gives in her papers, *Kristeva Reader*. The Bakhtinian *word* is a general category; it could have many interpretations because it is for its own nature ambiguous. Kristeva considers it as a spatial *unity* of the text. I do not share the same view but I think that it could be a short way to end my argument. Considering this spatial unity of the literary text as my starting point would take as a consequence

² C.S. Peirce, *Peirce on Signs: Writings on Semiotic*, University of North Carolina Press, Chapel Hill, NC, 1991. P239

that the object oriented paradigm could be a style of writing text with an abstract formalization, but what the programming languages would end up sharing with literature is only the fact that they occupy a formal space, this obscure *word* she describes. I think that this interpretation is reductive in literary terms, even if it is socially and historically contextualized, and shares only parts with the horizon on literary meaning and literary languages that Bakhtin has delineated speaking about literary *unity*.

The Bakhtinian word is a structure based on meaning, but what is under consideration when he speaks about it is the linguistic form, the meaning of it, and besides the form of the meaning itself, inside the context represented in the text. This context is not only a space, but it is a relation between time, space and narrated action, called by Bakhtin *Chronotope*. The *Chronotope* is used, in the Bakhtinian treatise, to identify the meaning of the novel. Bakhtin takes different literary forms under consideration to show how they organize the relation between space and time in the narration and to deduct from it, as a category, the *image of the man* expressed in the novelistic literary text. The result is a clear image of characters and their abilities, a range of languages uses that goes from more recognizable literary structures to the ambiguous one and dialogical one. A classical instance of this *chronotope* is the *time of adventure*, that defines a specific relation between space and time through a large range of possible forms in the text.

A typical adventure-time takes place in an ordinary context that can change in the different literary forms that Bakhtin analyses. In this type of time the rules and life are ordinarily following the shared social sense of the culture they belong to. Then *suddenly* something happens and breaks this equilibrium. What happens is always *extraordinary* and takes all characters out of their limb into a different type of time. In this different type of time events are never really caused by the characters, the actions of the character are more characteristic of reactions. The adventure time starts with a supernatural event; there could be monsters, or ghosts, or something similarly fictional. The reactions of the character are made on the purpose to reaffirm the rules that have been broken, and they usually represent a proof of nobility, for the hero or protagonist. The attainment of those proofs stops the course of adventure time, renewing the rules of the previous ordinary time, and submitting again to the rules of normal reality.

Looking at this narrative structure I could define a character as a variable. I can state it to change, as I prefer, during the execution, till the end, where it will change back to what it was at beginning of the execution. A typical instance of the adventure time is *Greek-romance*, in Bakhtin papers. And this is exactly how he defines the adventure time in this specific cultural and historical context:

"The Greek romance utilized and focused together in its structure almost all genres of ancient literature."³

The Iliad, for instance, begins with an event that happens to the character of Achilles. But you could say that it begins because of it. When the "wrath of Achilles" ends, the story finishes. Or to be more precise, when Achilles ends, as a sort of variable, then the story finishes. You can easily translate this in a logical perspective that includes all the narrated events of Iliad in a "While Wrath= True" statement. The Iliad does not end with the victory of the Greeks but with Achilles death. The story of the fall of the city is not included in the Iliad, because the poem is written in a static form and it is made to be a statue to one kind of literary language. Of course from the Iliad a lot of stories start. Odyssey for example, even the story of Orestes, used in the tragedy. But the type of time that is

³ M.Bakhtin, *The dialogical imagination*, Austin, 1987, P89

used in the epic novel is described by Bakhtin as the *time of the fathers*, so it is not based only on the common sense of rule in Greek society, it includes a shared memory of the past.

The time of adventure is naturally still used today, but it is not any more so strictly respected. Contemporary literature does not need to be so statuary when telling a story, because we have many stable literary support in which language is defined and refined. Besides, our standard "character" today is more active and surprising than any "statuary" image of old epic poems, because the image of the man we get from it is really limited and passive. Our contemporary literary approach is a stylization of older, rigid *chronotopes*; we "use" them in literature. We represent them because we want to discuss the principles that they state. This is the duplicity of the Bakhtinian *word*: a stylization of a formal language as the presence of a double meaning. The form is used to represent something else that is not included in the first level of signification.

It is intuitive that the chronotope identified by Bakhtin in the literary text, and in particular adventure time in its abstraction, reminds the *flow of execution* which similarly manages a number of formalized actions in a given time, coming from outside the text. This is perhaps an exhaustive structure to compare software text and literary text. The definition of a structure of the flow of execution in relation to some formalized concepts of software could be found in many programming paradigms, like Procedural and Structured programming. It is not by chance that a notion of a literate development of software had already been developed during the 90's, and was called "literate programming".

Literate programming is a term used by Donald E. Knuth, in the introduction of the WEB system in 1992⁴. The purpose of his work is to write a text for his software that could be both software text and the explanation of it, through a scheme made by macros. Those macros are contextual indexes based on the concepts of the programming language they refer to (variables, definitions of functions, etc.). The text of the software in WEB language has to be processed by a PASCAL interpreter, to generate the executable file. The macros of the web file can be processed through a TEX engine. The result will be a .tex file of documentation about how each program section works and why it was written as it is. So the file has to be processed more generally through a programming language and a document formatting language. All the information needed to be written in the source code following a specific syntax, the one of the WEB system, that is described by Knuth as follows:

"The main point is that the WEB is inherently bilingual, and that such a combination of languages proves to be much more powerful than either single language itself WEB does not make the other languages obsolete; on the contrary it enhances them."⁵

A section of the web software starts for instance with comments and ends with the program code they refer to. Between them there is an index indication, the macro, included in identification characters. And at the end of the program WEB there is an index of these macros and of the section where they appear. This WEB text is described as bilingual, but it is because it organizes the text into different parts one in a formal language and the second in an informal and human readable language. Knuth developed such a style of writing software more for its practical use (less debugging time, more detailed documentation on the work already done) than for attributing any kind of literary status to the software. If you are used to literary text maybe the text of the WEB file will seem to you like a paraphrases of a poem, it is the explanation of something enclosed in the form,

⁴ Knuth, Donald E. *Literate Programming* 1984 CSLI 1992

⁵ Ibid. P2

like a thought, or a process. Only the text of poetry has a comparable enclosure for interpretation and for the definition of its meaning that is based on its form, and anyway has to be explained in "Informal language" or prose.

Bakhtin proposes one of the most famous definitions of the difference between poetic and prose, in the same papers, *The dialogic Imagination*, where he speaks about *word* and *chronotope*. He takes as his model of the poetic the Epic, and as the model of the prose the novel. He expresses the difference between those two kinds of word in terms of time. So the statutory language used in the Epic is the one I have already described for the Iliad, and it is made in strict formal rules, as an attempt to build a unified idea of language, a harmonic example of it. On the contrary the prose word is the more stylized and the most structured in the level of meaning. The stylized word is the one we use in the contemporary novel, even if its specific kind of signification, as Bakhtin describes, takes its origin from ancient historiography.

Artificial Memories

In the overview about Object orientation I quoted B. Viola's article, and in particular the connection between Data-Space and the rhetoric theory of "loci" he profiles in it. The treatise on artificial memory in the rhetoric historical period he mentions is one of the most complete and it does not only speak about memory and spaces. It instead describes a contraposition between what was called "memory of things" and what was called "memory of words" by rhetoricians. In *The art of memory* F.A.Yates sustains that there is not enough documentation about the memory of word, which is only mentioned by persons like Cicero. It is defined generally in the most ancient treatise on rhetoric: *Ad Herennium*. The problem is that it is described only in terms of practice (The treatise was written for educative purpose), so the definition of memory of "loci" seems inexplicably useful to define two kinds of mnemonic techniques, the memory of words and the memory of things. The second part of the following definition describes this connection:

"The art of memory is like an inner writing. Those who know the letters of the alphabet can write down what is dictated to them and read out what they have written. Likewise those who have learned mnemonics can set in places what they have heard and deliver it from memory. "For the places are very much like wax tables and papyrus, the images are the letters, the arrangement and the disposition of the images like the script, and the delivery is like the reading."

Rhetoricians were literally writing images in their memories, through repetition:

"The ability to use these [images] will be supplied by practice which engenders habit, and [by images] of similar words changed and unchanged in case of drawn [from denoting] the part to denoting the genus, and by using the image of one word to remind of a whole sentence, as a consummate painter distinguishing the position of objects by modifying their shapes."⁶

Viola considers only the memory of things described in F.Yates treatise as the one that has had more success in the rhetorical practice. However the memory of things has not the formal requirement to be compared to the written form of an object oriented programming languages, because it is mostly based on psychological dynamics. The "memory of words" instead should formally work in the same way as a digital memory.

⁶ F.A.Yates, *The Art of memory*, Pimlico, London 1992 P33

Yates gave me others possibility of defining this memory of words, because she is very sensitive to formal and literary matters, in her treatise, so she searches for something enough old to be contemporary with the memory of words' culture. Yates finds a text taken from *Phaedrus* dialogue, which speaks about memory and writing. Behind every philosophical concept expressed in those lines, what is important here is an attitude that Socrates gives in relation to written text in the same dialogue. But before considering it we should know that who is speaking here is not Socrates himself (Socrates was against writing, especially if the subject was philosophy). He did not write anything but Plato did. So the only documentation that we have about him is what Plato wrote. Therefore if in *Phaedrus* is written "Socrates", we should take the chance to doubt it.

So "whoever" in this dialogue speaks with Socrates names, he is in the situation of reading a rhetorical discourse proposed by a student (from which the dialogue takes name, *Phaedrus*). This discourse is written by a sophist and speaks about an argument, comparing benefits and damages of it, and then tracing a perspective on it. The "instance" of Socrates sustains that the discourse is actually meaningless; he describes clearly the practice of sophist mnemonic techniques as a practice, with his usual moral style:

"Phaedrus: What do you mean, my good Socrates? How can you imagine that my unpractised memory can do justice to an elaborate work, which the greatest rhetorician of the age spent a long time composing? Indeed, I cannot; I would give a great deal if I could.

Socrates: I believe that I know Phaedrus about as well as I know myself, and I am very sure that the speech of Lysias was repeated to him, not once only, but again and again; -he insisted on hearing it many times over and Lysias was very willing to gratify him; at last, when nothing else would do, he got hold of the book, and looked at what he most wanted to see, - this occupied him during the whole morning; -and then when he was tired with sitting, he went out to take a walk, not until, by the dog, as I believe, he had simply learned by heart the entire discourse, unless it was unusually long, and he went to a place outside the wall that he might practise his lesson. there he saw a certain lover of discourse who had a similar weakness; - he saw and rejoiced; now though he, 'I shall have a partner in my revels.' And he invited him to come and walk with him. But when the lover of discourse begged that he would repeat the tale, he gave himself airs and said, 'No I cannot', as if he were indisposed; although, if the hearer had refused, he would sooner or later have been compelled by him to listen whether he would or no. Therefore Phaedrus, bid him do at once what he will so on do whether bidden or not."

This discourse is static. you cannot ask to it what does it mean, and it cannot explain itself anyway. The words remains at level of statement.

"Phaedrus: I have never noticed it; but I beseech you to tell me, Socrates, do you believe this tale?

Socrates: The wise are doubtful, and I should not be singular if, like them, I too doubted. I might have a rational explanation that Orithyia was playing with Pharmacia, when a northern gust carried her over the neighbouring rocks; and this being the manner of her death, she was said to have been carried away by Boreas. There is a discrepancy, however, about the locality: according to another version of the story she was taken from areophagus, and not from this place. Now I quite acknowledge that these allegories are very nice, but he is not to be envied who has to invent them: much labour and ingenuity will be required of him; and when he has once begun, he must go on and rehabilitate Hippocentaurs and chimeras dire. Gorgons and winged steeds flow in apace, and numberless other inconceivable and portentous natures. And if he is sceptical about them, and

would fail to reduce them one after another to the rules of probability, this sort of crude philosophy will take up a great deal of time."

We could consider for example that what Socrates says about the rhetoric text he had to comment is not at all new to us. I mean, the fact that the text he was reading was, or not, an exponent of the rhetorical memory of word products, that is quite likely: he expresses an uncomfortable sensation of the "dialectic though", a sort of paralysis in front of a written text that presents a refined poetic. He says that this poetic could be reduced to probabilities of truth. It is natural to wonder if this kind of formal obscurity, in rhetorical terms, is applicable to the transparency of the interface in software design as well and specifically to Object oriented interfacing features.

But the point here is not to measure the capacity of object oriented programming languages to define moral discourses, it is that they are inscribed in a memory, just like memory of words. And they both state a sort of "sequence of characters" inside two different kind of memories, without a real interpretation, but through a calculation in programming languages and repetition in memory of words. And finally they are both able to produce images manipulation, one inside a software context and the other inside a literary one through allegoric images.

The right formulas to compare

All the arguments I explained till now have a useless side; they are too general to describe the technical specificity of oral poetics. We have spoken about historically determined cases of Epic: the Greek Homeric poem and the rhetoric tradition. But it is not a formal definition of the oral poetic form; it is an explanation of procedures to learn it, following ancient rules of education that are not in discussion now. The description needed in a treatise of Narratology, "*The nature of narrative*" that includes the observation of a researcher that used to make real experiments with an oral poetic tradition to study how the oral poetic was related to human memory.

"Starting with the written text of the Homeric poems as they have come down to us Parry noticed that the traditional epithets and locutions, which have to a minor extent always constituted an element of the, "epic style" in subsequent Western tradition, were invariably used by Homer in the same metrical and semantic situations. These traditional elements of the Homeric dictions, in quantity and quality vastly richer than in the works of later poets, he called formulas. He defined the formula as "a group of words, which is regularly employed under the same metrical conditions to express a given essential idea.". Such fixed epithets as "son of Atreus" and "king of men" for Agamemnon, or "of the glancing helmet" for Hector, or "wine-dark", "loudly resounding" and "echoing" for the sea, have always been recognized as characteristic of Homer style, and their effect has been imitated by writers of literary epics from Apollonius onward. However, not until Parry discovered that the whole Homeric corpus, about 27,000 hexameter lines, was entirely formulaic did critics realize that what had long appeared to be only a superficial stylistic feature was in fact inescapable evidence that the Iliad and Odyssey were orally composed"⁷

The Homeric poems, as in all poetic forms that have an oral origin and composition, have a sort of formulaic structure. This structure is constituted by the repetition of specific expressions that refers to what Kristeva has described as linguistic and symbolic system. I mean, as is written in the

⁷ R. Sholes, R. Kellomaki, *The nature of narrative*, London: Oxford University Press, 1968, P 20

description of *Ad Herennium*, it is a sort of codifying of the "symbolic system", that is basically allegorical, inside a poetic language form. Parry called this system "formulaic". These formulas could be considered as the formal definitions of the different statements of the "allegoric images" used by the sophist to construct their discourses. Could this be the essence of the memory of words? Anyway the observations of Parry went on and he tried to understand how those "formulas" were related to human memory in the practice, this is a short description of the experiment and of Parry's conclusion:

"The singers themselves think that they are capable of repeating a whole epic verbatim, and take pride in their memory of what they must conceive to be a kind of fixed 'oral text'. When Parry took down the same song twice from the same singer, however, he discovered that exact correspondences between two performances were rare. Individual lines and episodes were composed differently in the two versions, but they both used the same formulas."⁸

I would keep the focus on this concept of oral text, because it is related to the level of execution of the poem and the level of memory. The oral text seems to be "what" is inscribed in human memory, and what generates the performance of the poet.

Object oriented word

The dynamic previously described is reproduced in OO poetic.cgi where every python command is defined as a formula, describing the mathematical and logical contents of it and the operations it is supposed to perform. The content of those formulas is developed in a wiki, which is The Object Oriented Poetry wiki, and which is connected to the cgi through the web API of the CMS Mediawiki, used to build the wiki website.

Bibliography

Bakhtin M. *The dialogical Imagination*, Austin 1987

Deleuze G. *Cinema 2, The time-image*, Athlone Press, London, 2000

Knuth D., *Literate programming*, 1984 CSLI 1992

Kristeva J. *The Kristeva Reader*, Columbia University press 1986

Peirce C.S. *Peirce on Signs: Writings on Semiotic*, University of North Carolina press, Chapel Hill, NC 1991

Plato. *Phaedrus*, translator: B.Jowett, project Gutenberg 2008

Scholes R., Kellogg R. *The nature of narrative*, London: Oxford University press, 1968

Wardrip-Fruin N., Montfort N., *The new media reader*, Cambridge, MIT press 2003

Yates F.A., *The art of memory*, Pimlico, London 1992

⁸ Ibid. P 25