The Functional Point of View: New Artistic Forms for Programmed Literary Works

Philippe Bootz

This paper examines the consequences of a new point of view about computer literature (termed the functional point of view) in which the concept of "text" is understood mainly as a process and not only as the object the reader sees on screen. To modelize—in the physical meaning of this term—how this new reality of text works, this paper describes the communication process between the author and the reader in a formalism (the functional model) that brings out the functions (such as writing or reading) in this process (Fig. 1 encapsulates the results of the model in the form of a chart). In this process, a particular type of work appears: procedural texts, in which a nontraditional function, termed "generation," takes place in between writing and reading. Presently, only programmed computer works can be procedural works, but this could change in the future because "procedural texts" is a concept and not a technical characteristic. This paper examines procedural works only and invites us to consider the concept of a text consisting of several objects (and not a single object, as in the traditional concept of text) and a function.

The complete description (encapsulated in Fig. 2) of this new concept of text (named the "textual process" to emphasize the notion of process) requires a description of the running of the generation function and a description of the relationships between the various textual objects in the model. This description shows the emergence of unexpected behavior in a computer work, which makes new poetic forms possible. This paper will not develop the complete description of the model. It will only define useful terms and explain some of these possible new literary forms (unique-reading poem, locked works and adaptive generators), which invoke the sense of the "tragic" in literature, in which no one (neither the author nor the reader) controls the textual process. The first of these forms, the unique-reading poem, has been realized in the poem "passage" published in 1997 [1]. The locked works will emerge in the near future (the functional point of view serves as a pilot program for this poetic form). To understand how this new point of view historically appears, I will first examine the history of computer literature in Europe and the present literary forms of computer literature.

HISTORY AND CURRENT FORMS

Computer literature explores several paths. Some present-day productions, specially on the Web, are collective writings, works in progress. But others, generally published on CD-ROM, even if some of them can be read on the Web, are more traditional communication processes between a single author and a single reader. All these texts can be traced back...
to three basic forms: hypertext, automatic generators and animated texts. Hypertext is the most popular of these forms in the U.S.A., but it is not the most important in Europe. These three forms were translated to the computer during the same years: 1978–1985. But, even if the computer accentuates the possibilities of these forms, none of them were originally created on computer. Hypertexts, generators and animated texts can be made without computers, but the computer is the truly adapted machine for these forms.

In Europe, the first important movement was A.L.A.M.O. (Atelier de Litterature Assistee par la Mathematique et les Ordinateur—Workshop of Literature Assisted by Mathematics and Computers), which created the automatic generator form. The most important authors using this form are Jean-Pierre Balpe (from A.L.A.M.O.) and Bernard Magne. Many people in Europe presently use this form (especially in Italy and Portugal). The automatic generator, based on the generative grammar of Chomsky, creates texts from a dictionary and grammar tools. Texts created by computer are not written by the author of the program. Jean-Pierre Balpe uses this technique to produce infinite novels. Automatic generators can also be regarded as a combinatorial process, in the way of the “100.000 milliards de poèmes” by Raymond Queneau. Some automatic generators can be run on the Web (see, for instance, the site of infolipo in Geneva <http://www.unige.ch/infolipo>). The automatic generator movement is a very important movement in Europe, much as hypertext is in the U.S.A., since many authors use this technique (however, unlike hypertext, there is no corresponding software such as Storyspace by Eastgate System), and it raises some questions relevant to literature: What is really a text? Can a text have no author? What does it mean to read a particular text amongst an infinity of possible texts? etc. Generally, until 1989, generated texts were printed out by a printing device and were read like classical texts.

Animated texts were created on computer by the team L.A.I.R.E. (Lecture, Art, Innovation, Recherche, Ecriture—Reading, Art, Innovation, Research, Writing) composed of Philippe Bootz, Frédéric Develay, Jean-Marie Dutey, Claude Maillard and Tibor Papp. All these authors came from concrete, sound and visual poetry. Animated poetry introduced time within the written text, as part of it. This fact breaks down important classical features of a text, such as the concepts of a sentence or a work. It creates a continuum between text and not-text, a transition between texts.

L.A.I.R.E. published, for the first time, an electronic journal of poetry, alire, in January 1989, which included animated texts on floppy disks but also hypertexts and automatic generators. This review is certainly the oldest multimedia literary journal. It is presently published by MOTS-VOIR in Villeneuve d’Ascq (France) and the last issue, alire10/DOC(K)S, was published on CD-ROM in conjunction with the journal DOC(K)S, published by AKEATON (France). Alire had a prominent influence and developed computer literature for private reading before the boom of multimedia and of the Web. Because generators, animated texts and unclassical hypertexts were published in alire, alire presents us with three characteristics of computer literature. The first is the similarity between the three basic forms when all of them are seen on a screen: in programmed literature, all three forms use the same lines of code. All of them produce non-permanent texts, for which time is a very important dimension. The second, which breaks with the traditional behavior of texts, is the importance of the diachronic behavior of the works. Because of the quick evolution of the performance of machines, the programs do not give exactly the same result on screen from one reading to another, even if the program itself is not interactive. This particularity, typical of computer media and programmed works, appears only with time, because the CD-ROM is a permanent archive.

This fact incites us to think that literature is not only in the “result” of the program, i.e. the text (in a classical meaning) that appears on the screen (named the “text-to-be-seen” in this paper), but also, and more importantly, the process that goes on by itself to produce this text-to-be-seen. This point of view is called the “functional point of view” in this paper because it focuses not on the text-to-be-seen (which appears as a literary work in the traditional meaning), but on the process that is going on, which can be described as a mathematical function that acts on some material (program, data) to produce the text-to-be-seen. This point of view has two important consequences. The first is a change in the understanding of what computer art actually is, and the second is the possibility of a new approach to literature and more generally the foreseeable emergence of new artistic forms based on the particularity of this process, possible only in the computer arts and without any classical equivalent.

Some of these new forms are “the unique-reading poem,” developed in the work “passage” published in alire10, and the “locked works.” The unique-reading form is a hybrid form between the automatic generator and hypertext that produces a personal text-to-be-seen that grows like a flower, through interactivity. In a unique-reading poem, the text-to-be-seen stems irrevocably from both the author and the reader. The “locked works” form uses some of the possibilities inherent in the computing process to free the chronology of reading from the chronology of writing. In traditional literature, the end of a text cannot be read before the beginning, and multi-volume books (a trilogy, for instance) have to be read in the proper order, lest the coherence be affected. The locked works break with this. The coherence of the reading be-

![Fig. 2. The textual process (detail of the functional pattern) consists of the generation function and its input and output textual objects.](image-url)

---

308 *Bootz, The Functional Point of View*
comes independent of the chronology of the writing, even if works that are read are strongly tied together. This very particular process is only possible by use of an automatic generator. The locked works can be described thus: “Any action of a reader on poem A will affect in a coherent manner the possible texts-to-be-seen produced by another poem B, which will be read after A, even if B was created by the author 20 years before A.” It is obvious in this sentence that the author does not know what A will be when creating B. This super-structure “locks” works A and B by the act of reading. If all products of a writer are overlaid with this super-structure, it can be asserted that every reader will construct his or her “private author” by reading, and the literary knowledge about an author will become reading-dependent and no longer works-dependent.

In the new understanding of programmed computer art, it appears that the author cannot generally obtain all the elements he would like to have on the screen of the reader’s computer. The author does not control all the useful parameters of the computer of the reader and thus the author’s activity changes: He no longer creates a new world; instead he has to manage the inability of a computer to generate this world. This is the “tragic” point of view in the Greek sense: the author is not the master of his fate as author. But it is not a pessimistic point of view because this “tragedy” is balanced by another power if he accepts this fact. Technically, the algorithms he can use to “manage” this gap are able to program works that run on various machines and produce very different texts-to-be-seen. So, any reader can say “I have read this poem.” With this tool (named an adaptive generator, which is actually in construction) the poem will “live” in machines, as plants do when transplanted to a new pot.

This paper explains how all these particularities appear in a computer poem. To do this, I will explore the total process of communication between the author and the reader as a system. I use a particular theoretical approach called a “systemic analysis” to study this system. The approach can be described in two complementary ways: a structural analysis, which describes the material elements of the system (for instance the computer), and a functional analysis, which extracts from the elements those characteristics necessary to study the functions made by them (such as generating or reading). This second way is what I have used in this paper because it focuses on the process itself. This paper was first published in *alire10* (which contains a book and a CD-ROM; theoretical papers are printed in the book, and the CD-ROM contains the computer literary works). The original framework was developed in 1996, in issue No. 8 of the review *Les Cahiers du CIRCAV* [2].

**AND SO IT GOES, BUT WHAT FOR?**

The act of reading the poem only once dates back to the invention of the automatic generator. Does it not assert by practice the fact that the poem basically is not meant to be read, or in any case, not in the way our bookish culture has forced reading on us? Reading only once—or else possibly transforming re-reading into a permanent quest. While reinventing reading, the poet has, once again, freed him- or herself from everything possible in order to embark, mind and heart, into the unexpected, which is unforgivable in our society. Reading becomes an unpredictable experience for each person according to his or her mouse moves. It is the text-pretext-to-the-text of A. This very different texts-to-be-seen. So, any person, because there exists a community of readers who, while reading what another person is not reading, make that person’s reading possible through program exchanges. It is also the “locked works” and the unique-reading poem, where reading builds up its own possibility in time in the same way that humanity builds up history as a dialogue between hope and constraint.

While experiencing poem modes that could not be read in their entirety, poetry has entered a new phase: having readers read not what is not readable, but rather experience the significance of that which is not readable.

How? and what for? It is to this prospect that I am inviting the readers of this article, which will be limited to the description of an approach. It is by first perceiving and describing the way a programmed text functions that the different aspects of the approach will be explained.

**THE FUNCTIONAL MODEL**

**The General Pattern**

Functional analysis consists of describing in terms of functionalities rather than objects. It brings to the fore the active functions of a system, as well as the entities upon which these functions act and those they create. The physical elements that allow one to realize these functions are not taken into account, or are not considered by the model, and this requires a structural analysis of the system, as a complement to the functional analysis. We shall not go as far as the latter, which is inefficient, in order to deal with the aspects that are of interest to us here. It is clear, however, that poor author-programmers such as myself are obliged to go through this to bring our realizations to satisfactory completion.

Functional analysis is a descriptive tool, not to be mistaken with the “functional point of view,” which I will discuss later on. I propose a model here to describe the complete communication chain (called the “work” later on) between an author, defined as the initiating subject of the communication process, and a reader, defined as the targeted subject of this process.

Figure 1 shows a general idea of the properties that are useful to our purpose [7]. In this model, “writing” and “reading” do not coincide exactly with the operations meant by these words in common usage. For instance, the writing function can be partly realized by a person (called a “realizer” in my model [8]) other than the author. But the main difference comes from the existence of a new function, that of “generation,” which is inserted between “reading” and “writing” and which in computer literature is largely performed by the computer. This function does not limit itself to what is usually called in computer literature “automatic text generation.”

The names given to these functions were chosen to make the model in Fig. 1 self-explanatory, but the functions are defined only inside the model, by the actions they perform upon the working materials that circulate between them (as shown above and below the arrows in Fig. 1 and not by any outside subjectivity)
The Textual Process
It is a common habit to speak of the text as an object, something that could have its own existence, possibly even outside the medium. Such a text can then be “exported” to any of several media: book, video, floppy disk, etc. This may be true of a large number of texts, but not those that I am talking about here. What is called the “text’s field” in the model is not an object but a complete subsystem (Fig. 2). Now, a process does not exist statically and does not fit in with any medium. One can only describe it and make it tangible by activating it. This is very precisely where the “immaterial” nature of computer literature lies—not in the fact that the object is accessible to the reader merely as transitory light or a series of bits inside a memory. Compared to the immaterial nature of the textual process, a luminous sculpture is totally material.

The fact that such a model should substantially differ from the classical literature model leads to the idea that there exist an infinite number of possible types of literature that are all different in their functional patterns. One could imagine given literature models, then determine which literary worlds would realize these models—a meta oulipo [9] at the literature level rather than at the text level, so to speak. But let the reader be reassured, I shall deal in this paper only with programmed texts read on computer.

The Characteristics of the Work in the Functional Model
The functions of reading and writing are completed by the author (even if a team is concerned) and the reader (an individual or a group). The very existence of a new function inserted between reading and writing creates a separation, or disconnection, between the author and the reader, which is taken advantage of by the author in interactive productions. The main consequence [10] of this separation is to render the author’s real work—and, therefore, the whole of the work (considered here as the special production that the reader receives and not the whole of an author’s productions)—impossible to know. No one can assert “having read” the text. This separation classically causes the object “text” to explode into a number of “textual objects” whose physical access by the reader is no longer a sufficient condition for reading. These objects appear in different places of the model as functional material and preserve only a few characteristics of each object that would allow it to be called “text” in the classical sense.

These objects are not always present in the same place or at the same time in the communication process described by the model, and their collection does not therefore reproduce the usual notion of a text, in which the text is understood to be an object.

The main textual objects in the model are the following:

The Written-Text: The written-text (shown in Fig. 3 [11]) corresponds to the author’s project before any description is given of it. It is the most abstract textual object, expressed only through the multiple descriptions that the author may make of it: discursive, symbolic, graphic or other, which may or may not be designed for a computer. It is this object that makes up the primary creation by the author, and no one else intervenes; this accounts for its name.

The Author-Text: Completed by the writing function, the author-text describes the elements of the written text that are necessary for the generation of terms that are understandable both by the author and by the agent of the generation function (the computer, in the case of the texts we are concerned with here). This author-text is formed by two main classes of objects, named according to their computer: the source, which is an ordered structure of orders (this is the main element of the author-text), and the data, a series of materials whose structures are set forms but are not the contents. (Source and data appear in Fig. 3.) The whole of the author-texts cannot be reached by the reader.

The Text-To-Be-Seen: This is the object to which the reader can have access. It is spatiotemporal, such as an animated text, and attached to the medium of the screen. The text-to-be-seen is, from the reader’s point of view, that which is most like the traditional text object. As a matter of fact, it is the only object that, for the reader, can play this role. It can offer the deceptive appearance of a non-procedural form of literature (that is, whose functional description does not include the generation function). The author, on the contrary, cannot have access to this text-to-be-seen. One can spot, in the text-to-be-seen, whatever it may be, sentence-texts, defined as text objects that the reader would obtain by copying the whole of the sentences that are offered in the text-to-be-seen. In no case whatsoever can the characteristics of a “text on computer” be reduced to those of the sentence-texts, as far as the work’s meaning or style or literary nature is concerned. Even though the existence of these sentence-texts makes computer literature differ from other computer arts (the proposed functional model actually makes it possible to describe a wide range of computer arts, with a few adjustments), the work that generates them does not limit itself to these sentence-texts. An author who transfers a text object designed for paper onto the screen without transforming it is only writing “on” the computer. As a matter of fact, one can note that the words in the text-to-be-seen can be organized into several different sentence-texts, especially in the case of animated texts-to-be-seen; this contributes to the multiple meanings of the texts-to-be-seen.

The Read-Text: This is the mental representation of the work (as a result of the complete communication process, and not merely the text-to-be-seen) that the reader brings into being. This separation between reader and author [12] is accompanied by the relative autonomy of the textual process, which, since it is realized by the computer, to a large extent evades the author’s control as well as the reader’s, without being the work of an electronic deus ex machina. Later in this paper I shall detail the model’s elements that describe this autonomy. It is one of the reasons why, in my aesthetic approach “what is written is not designed for reading.”

This position (one of the bases of the functional point of view), expressed here in a paradoxical manner, is not a denial of text or the last refuge of modesty, but an acceptance of the way these procedural works function, which requires compromise on the part of author and reader. It is much better to conceive the work without intending to dominate all possible characteristics of the text-to-be-seen, since some of the characteristics might very well not ever be realized or might be altered during its generation [13].

The last characteristic that I shall mention about the generation function is its interaction with reading. The reader cannot achieve the reading stage without bringing the generation function into play, because the works that are to be read are read in allire on the reader’s computer [14], which will likely differ from the author’s. This situation
can have a notable impact on the text-to-be-seen. This interaction is functional and fundamentally not linked to the interactive or non-interactive nature of the work [15]. (This phenomenon compelled me, in 1994, to rewrite all of the first issues of \textit{alire}, allowing an author to “guide” the reading mode in order to attempt to manage the activation of the textual process by the reader.)

Thus, the author’s work, in order to manage the textual process, should enable the author to create in the author-texts the only material that he or she can have access to—the software components and the data that are likely, on the one hand, to guide the implementation of the textual process by the reader (not like a “help” file that guides one in a software program, but like a shepherd guiding the herd to its intended destination), and, on the other hand, to tame (as much as possible!) the autonomy of the textual process.

**THE FUNCTIONAL POINT OF VIEW**

The first goal (guiding the textual process) led me to conceive the unique-reading poem, whose first version is to be found in the CD-ROM of \textit{alire10} [16] and in the body of my locked works, which is presently being programmed. The options I have chosen in these projects can, to a large extent, be detected by the reader.

The second goal, taming the autonomy of the textual process, leads to the idea of an adaptive generator whose characteristics become apparent to the reader as its material configuration gathers momentum. Presenting, within the framework of the functional model, the impact of these diverse “writings” upon the works created (and not merely on the generated text) requires presenting the functional point of view, an author’s ideological position about the work’s ultimate aim, outside the functional model.

**The Need for the Author’s Commitment**

Early in this article, I mentioned that one of the fundamental choices was whether the author wishes to give a mimetic or non-mimetic nature to the generated texts-to-be-seen. Separation is actually the first characteristic of a procedural text on computer that the author is faced with when placing the work at a reader’s disposal. The author, therefore, has to take a position on the management of this aspect, even if only through the choice of the tools of creation. We are then the witnesses of a commitment, purely internal to literature, on the part of the author. If there is a commitment on the part of the author, then he or she is rightly entitled to expect the reader’s commitment in return. (This has been a constant position in \textit{alire} since its creation.)

Yet to understand the nature of the commitment the author is to achieve, we somehow have to detail the elements that limit the procedural text’s “faithfulness” and the clues that make it possible to qualify, if not measure, the readability of a procedural text. We shall then be able to specify the implications of this commitment in the author’s work, as I have just described.

**The Text-To-Be-Seen’s Readability and the Reading Process**

Readability is a characteristic of texts-to-be-seen. It can only be determined from the reader’s point of view. Faithfulness is an estimate of the appropriateness between a text-to-be-seen realized in an individual reading and that which would be obtained on the author’s computer in similar textual circumstances. Although this characteristic of the text-to-be-seen is not accessible to the reader, it is accessible to the “critic” (or observer), who realizes not a reading action, but the observation of a reading act. At the present stage of the development of computer literature, it is the author who, most of the time, plays the role of critic, during shows, performances or exhibitions, for instance.

Observing readers reading shows that a small number of characteristics of the textual process can ensure readability. It is enough if the text-to-be-seen “looks” like a non-procedural text—i.e. if it can be obtained by a simple “call” procedure (by a summary, for instance), a function similar to the opening of a book (conditioning, connecting operation), if it ends with an errorless closing of the program (disconnecting operation) and if between these two procedures two conditions are realized: (1) if sentence-texts can be built from elements in the text-to-be-seen (even if a large amount of information evades the reader’s sagacity) and (2) if no error message is generated. In other words, it is enough to ensure a minimum material compatibility between the orders of the author-text (concerning, among others, the operating system, the speed of execution and the material data, such as screen resolution and the number of colors), as well as a debugging of the program, to ensure the readability of the text-to-be-seen. The latter condition is preliminary to the reading, but its resemblance to the preliminary operations of reading a non-procedural text may constitute a
trap for the reader while letting him or her extrapolate other characteristics of the non-procedural text that do not necessarily apply to a computer work, particularly by letting the reader suppose that the first elements of the text-to-be-seen correspond to the opening words of the work and that the last ones are an end to it. This may not be true at all.

The generation has most likely already dealt with part of the source (the initializations phases of the generation process) before the first elements of the text-to-be-seen appear. In the same way, the generator generally realizes a certain number of operations outside the text-to-be-seen after the latter is completed. The text-to-be-seen does not constitute the whole of the generation result but only the part that is accessible to the reader (hence its name, which is not linked to its being or not being part of visual poetry, but to its characteristic of visibility by the reader).

The generation also produces the inferred data, which we shall come back to later and which make it possible, for instance, to prevent the reading in any specific order of the multimedia sequences in my unique-reading poem “passage,” even by direct call outside the menu, whereas all of the elements that are necessary to the activation of the sequences are physically on the CD-ROM. This non-central position of the text-to-be-seen in the textual process is reflected in the sentence-texts that come from it. The coherence of the text-to-be-seen, necessary for its reading, is linked to the need for a coherence, but not necessarily for this coherence. This is particularly the case with automatic generators. Yet, if the generator, as in the case of the unique-reading poem, generates only one text-to-be-seen, without the possibility of a reinitialization, the reader has no way of discerning the difference between the constant elements and those that are calculated inside what he or she is reading. The elements that influence the reading can then be mastered by the author and are part of managing the implementation of the textual process by the reader. Readability is not a guarantee, either, that the reader will read what the author has intended.

Thus, there have been a few meaningful experiences of textual processes being prematurely aborted, probably following a “wrong move” by the reader, which he or she did not spot and which, having generated no particular message, was not perceived as truncating the text-to-be-seen. The reader may read only an extract from the text-to-be-seen while thinking he or she has read the whole. The read-text which the reader has created has obviously been strongly influenced by this, as a reader’s remarks show [17].

Another example, even more extreme, is the possibility of a reader inadvertently activating a string of tasks that switches to another cultural product without the user being aware of it. This actually occurred during one exhibition [18], and the reader found himself outside the work while thinking he was going through it in an hypertextual way!

Generally the “unfaithfulness” linked to the autonomy of the textual process has, thankfully, more limited effects, such as, for instance, causing the loss of synchronization between events or reversing the source’s order during the execution: the sequence of events is not respected by their chronology.

**MANAGING THE READING CONTEXT: THE ADAPTIVE GENERATOR**

The first of these examples of unfaithfulness to the text-to-be-seen [19] illustrates the influence of the generation on the context in which the reader finds him- or herself immersed. Such scenarios are less likely to take place within the framework of an intimate and individual reading on the reader’s own computer than in the destabilizing and less-mastered framework of a public place. This difference illustrates the influence of the technical characteristics of individual computers, their speed among others. The likelihood of this influence is as strong at the reader’s home as it is in a public place. These two elements, the reading environment and the computer’s physical parameters, are combined in the model (see Fig. 3) under the generic term reading context and make up the component termed the reading data, which is independent of the reader. This reading context exists only during the reading and influences the way the textual process generates the text-to-be-seen. It is mastered neither by the author nor by the reader and is responsible for the autonomy of the textual process.

There is no need to modelize this reading context with something like a noise source. The notion of source (of a noise or a signal) is alien to the model, since such a source would be of a structural nature. What is more, even from a structural point of view, the notion of a source cannot be applied to the reading context, which does not actually consist of a disturbance that would be generated by a source outside the machine and superimposed on the signal dealt with by it, but consists rather of a coherent and ordered management that is allowed by the source but does not correspond to the author’s expectation. This is clearly a weakness in the description of the written-text by the author-text. Computer scientists in the industry are familiar with this problem, at least as far as the management of computers’ characteristics is concerned. This weakness
makes it necessary to take into account the fact that the execution of a program strongly depends on the machine that executes it, even if the realization of the program does not depend on the computer on which the source is written. It would therefore be wiser to talk of this performance in terms of the undefined influences in the author-text. Indeed, it is because the program does not specify, as a rule, the conditions and parameters necessary for the execution of the orders—and does not allow for a performance substitute in case these conditions are violated or these values are not respected—that the generation differs from the expected performance.

The influence of this reading context can be perceived in two opposite ways. One can consider that unfaithfulness corresponds to a violation of the written-text, and that the reading is wrong, possibly “illegal.” This is the point of view taken in the mimetic position. In order to avoid this awkward situation, the main characteristics of the machines on which the author-text have been written are mentioned. The functional point of view, which places greater importance on readability than on faithfulness, starts from the principle that whatever the reader may be reading, the reader is reading and that this reading is perfectly legitimate. In this perspective, reading failures and misinterpretation risks are strongly reduced. The text-to-be-seen can then accept contradictory readings without misinterpretation, something that is also seen in non-computer procedural visual productions.

While accepting, from the functional point of view, the autonomy of the textual process, the author can then alter his or her written-text so as to reconcile readability and faithfulness in the best way. More precisely, he or she needs to consider which characteristics are modifiable and to what extent. Respecting other characteristics is then no more than a “pleasant accident.” The author really has a commitment here, a total involvement in the work, but this involvement is pure, the author having left the concrete details of the text-to-be-seen in order to manage the conditions of its genesis. The author is, from the functional point of view, compelled to adapt his or her demands to the performances of the reader’s machine. The modes I have selected for this adaptation constitute the adaptive generator, now in the making. It is a protocol [20] (and not a work) derived from the techniques of managing industrial processes in real time, for this seems the most efficient method to manage the performance of a multitask system such as Windows. Being a protocol, the adaptive generator is a structural and not a functional construction, and it is linked to the tools used.

In order to finalize this protocol, from the functional point of view the text-to-be-seen is no longer considered an object, but a series of processes simultaneously at work [21]. For instance, an animated text-to-be-seen will not be worked upon as a film of 15 or 25 frames per second, but rather each action, each move, each change in color, etc., will be described and managed as an independent process designed to take place in real time. The author-text will then have to establish a protocol for the automatic management of the processes (how to create, activate, stop them; how to note their state and their results; how to determine duration, cycle time, etc.) and, most of all, include an authorizer—a program that, in industrial computer programming, manages the synchronization and conflicts of real-time processes. This authorizer should, among other things, be able to prioritize the processes, initialize the priorities, manage their synchronization, delete them. The authorizer’s work can only be achieved if the author sets up a request structure for the management of the processes (by defining priorities, intervals for cycle or total durations, rules for the dependence of processes) and if the generator determines, through a series of tests, the possibilities of the reader’s machine concerning each elementary process. Such an enterprise considerably increases the amount of programming work necessary, but the authorization has to be done only once per imagined type of process. In return, it guarantees readability on all the machines compatible with the language in which the authorizer is written and, with the coming of Java, possible compatibility with all platforms. It also guarantees faithfulness to an author’s project, which is no longer the genesis of a work but the management of its gaps. This management not only concerns the object the reader can have access to (the text-to-be-seen) but the whole textual process: computer writing cannot be reduced to the objectification of a project, unless the project itself is the project of a functioning process (to objectify the project of a functioning process is precisely to implement this functioning process). Formulating this in a different way will make the author’s tragic fall from paradise stand out more: in order to write he or she can no longer think about the final result, but only the writing. The author is therefore compelled to install the paths, i.e. the adaptive generator, that will enable him or her not to lose paradise too much.

But this loss of power to which the author agrees opens up unexplored realms; the functional point of view, filled with doubts and traps, is also the path to the adventure that assembles and builds. If the adaptive generator makes it possible to manage the gaps of a project that could not be activated to its fullest on the reader’s computer, it also makes it possible to render projects that are readable on all machines, even the less powerful ones. The example of the evolution over a long period (about 6 years) of a text such as Jean-Marie Dutey’s Le Mange-Textes [22] shows that works can be conceived that will yield as rich a performance, although different, on slow machines as on fast machines. The performance of the computer is not linked to the simplicity, the interest or the precision of a text-to-be-seen and does not therefore create “more” or a “better” result in the read-text. The computer can, however, introduce side-effects such as a change in the status of the work or possibly its complete metamorphosis. One can then conceive and realize a work, even one that is purely animated and non-interactive in the present sense of the term (“interactivity” defined as the reader’s conscious action), that slowly evolves over time in relation to the evolution of the reader’s equipment. The text-to-be-seen can be the realization of a constantly truncated project, and its reading that of a constantly postponed text.

A FIRST MANAGEMENT OF THE READING PROCESS: THE DIALOGUE IN THE UNIQUE-READING POEM

If the author’s work consists of managing the entire textual process, he or she has to guide, in addition to the readability, the implementation of the textual process by the reader. Other options from the functional point of view then come forward.

The main option is to give greater importance in the reading to the manifestation of the reader’s “experience” (erfahrung) in his or her legitimate quest for information and the acquisition of “knowledge” about the text (the text-to-
be-seen or written-text). This position is the opposite of our information society’s Pavlovian reflex. Procedural literature to me (and this is why it started being of interest to me long before I put it on the computer) is open warfare with journalistic reading. It gives back to the reader a place as an actor and a free subject, a living one at that, rather than merely being treated as a child to be guided or a glass to be filled, even if with beauty (which does not mean I do not also try to put beauty into computer poems). If there is to be a conflict between aesthetic pleasure and the experience of reading, necessarily irreversible and unbroken as it is, the text will always function so as to give greater importance to experience—by which I mean, among other things, the reader’s awareness that he or she is reading and that what is being read is going by, whether the reader catches it or not. The unique-reading poem takes this functioning very far, since it involves irreversibility and no re-reading. Due to this, it has significant demand for the reader as for the author of the poem: he or she is reading and that what is being read is going by, by which I mean, among other things, that the reader’s awareness that he or she is reading and that what is being read is going by, whether the reader catches it or not.

The unique-reading poem takes this functioning very far, since it involves irreversibility and no re-reading. Due to this, it has significant demand for the reader as for the author of the poem: he or she is reading and that what is being read is going by, by which I mean, among other things, that the reader’s awareness that he or she is reading and that what is being read is going by.

The functional point of view is as demanding for the reader as for the author. It imposes upon the reader the role of being an “actor” of his or her own reading and not merely a “spectator in hands and mind.” The wonderful freedom that is given to the reader goes along with a constraint that is just as important: it is impossible for the reader not to use this freedom (unless he or she loses it in the unique-reading poem). But isn’t that a symbolic representation of the conditions of democracy itself?

The predominance of the erfahrung over the deciphering brings about, depending on the level of management reached by the author, a certain number of consequences to the reading. The first is commonly a general feeling of frustration in readers who are approaching this kind of production for the first time. The unhappy defender ofwidowed and orphaned words is used to being able, in poems that are not unique-reading poems nor automatic generators, to fall back on rereading, which—except with automatic generators—makes it possible to revisit the polysemous relations (numerous as they are) of the text-to-be-seen according to the well-known games of the intellect. However, these relations rarely appear in their entirety in reading and rereading operations, since these relations are always implemented through a cultural background that inhibits some polysemous functionings and amplifies others. (It has, for instance, become a game for me to see how quickly I can discover whether a reader has a literary or arts education, through the way he or she reads or rereads an animated text, or a visual one. I am rarely mistaken in my guess.) Rereading actually performs another operation: it increases the coherence and wealth of the read-text by reinserting “memories” of the previous text-to-be-seen (which are in reality fragments of the read-text) into the text-to-be-seen in progress so as to “read it” [23] in a better way. The rereading is an attempt, a constructive one, to deny the fundamental temporality of the text-to-be-seen [24].

But the unique-reading poem goes further. Its structure is given in the comments one can have access to in the “passage” menu on the CD-ROM of alive10 [25]. Rereading “passage” means running the program again. But the program does not reset from one rereading to another. The rereading involves the reactivation of the same textual process and not the realization of the same text-to-be-seen. This process progresses in three phases. In the first phase, texts-to-be-seen are just animated, and rereading the poem is like reading the next chapter of a book (Fig. 4). In the second phase, the texts-to-be-seen are interactive. But the reader can never reset or cancel his actions, even if he restarts the computer (Color Plate B No. 2). The third phase is generated by use of the information given to the program by the reader in the second phase. Because this phase is not interactive, this information is the same at every rereading of this phase, and the generated text-to-be-seen is always the same. Thus, paradoxically, it seems to the reader that the poem is not unique-reading because he continues to read the same text-to-be-seen, while in fact the poem is truly unique-reading: another reader will read another text-to-be-seen, which will also be repetitive from one rereading to another in this third phase. The rereading, in the global process, is then part of the implementation of the final phase of the poem.

The participation of the reader is a necessary condition for the complete realization of the project, which leads to the canceling of the “unique reading” character of the last phase. The mastery of the meaning by the reader and the suitability of his or her choices will only have an influence on the degree of “truth” of the dialogue between the author and the reader, a dialogue in which the reader’s word provides the data for this interactive phase and the author’s answer provides it for the last phase. The interactive phase is a learning phase for the program, and the last phase is the author’s answer to the idea that he or she has (or, rather, that the program has, by proxy) of the reader. The dialogue is personalized; this is why the unique-reading poem establishes a discrimination between readers: setting the dialogue between the author and a particular reader supposes that all of the readings are performed by the same reader. The final text-to-be-seen then has a history—that of its “making”—for the reader who, little by little, manufactures it and for whom it is meant. This history is not for other readers, even if those readers themselves have also “read” “passage.” Let us note also and incidentally that “passage” uses the classic structures of computer literature in an uncommon way, which unsettles habits. Then the second phase, interactive as it is, looks very much like a generator “controlled” by the reader, whereas its structure is a hypertext with the particularities that it allows only one path per reader and that this path cannot go through the same nodes again. The nodes in this hypertext (which are made up of animated sequences) cannot be “explored”—that is, the reader cannot stop his or her action at these points in order to choose the next location. He or she can only go through them. The third phase merely looks like an animated poem, whereas it is actually an automatic generator that uses as a describer the induced data of the hypertext—an element of the functional model that we shall come back to below—so as to generate, here as well, only one text-to-be-seen per reader, identical at each rereading. On the whole, the performance of the text-to-be-seen does not reproduce the structure of the author-text, thus manifesting the independence of the author’s and the reader’s points of view.

**THE INFERRED DATA AND THE LOCKED WORKS**

The inferred data make up the part of the generated materials that is invisible to the reader during his or her reading. These materials correspond to the textual process and are in addition to those of the author. These two characteristics
place the inferred data in the author’s field, even if the reading is not alien to their development. It can then be considered that the machine plays a part of the writer’s role, which is, by tradition, allotted to the author in non-procedural literature. Should the computer be considered an artificial ghost-writer? Such an idea might more quickly come to mind when dealing with generators, for in this case the reader, having been invited to take a part in it, clearly identifies that the writing is not completed by the author. For me, the answer to such a question is a definite “No.” The induced data have all the characteristics of data—that is, only their content is generated; that is not the case for either the definition of their structure or for orders that might be added to the corpus of the source and would then expand its possibilities.

The fact that inferred data, reading data and the text-to-be-seen should be materials for the same function in no way signifies that they are simultaneous. In the same way, the intervention of the reader in the possible development of these data does not signify that the actions performed by him or her correspond to a reading in the model’s sense, since this character then does not play the role of the target of the process (a definition for the true reader), he or she is only an instrument [26] of it.

In my approach to a personalized dialogue, only the reader—at certain times and while remaining a reader—plays the role of an instrument, but for himself or herself alone. And he or she cannot but play this role. For instance, in the interactive phase of the unique-reading poem, the inferred data are the content of the descriptor used in the second phase. This temporal order of the states of the reader creates an axis of the times specific to reading, which is similar to the axis of narration in the oral tradition. In the African oral tradition, a story does not exist independently; it is reinvented at each narration and the chronology of the story is not that which was imposed by past narrators, who are multiple and anonymous anyway. A story exists at the time of the narration in progress, invented by the narrator. Such is what my “locked works” will produce: a chronology of the generators linked to the reading and not to the creation, a chronology for the exclusive use of the reader.

This chronology will function in the following way: a text 1, written before a text 2 by the author but read after it, will know the results of the reading of text 2 during its generation and will possibly be presented to the reader as having been written after text 2. That is, any of the reader’s actions in any text will direct the whole of the texts-to-be-seen that he or she will then read, whether these texts-to-be-seen were manufactured by works written before or after the one in which he or she is intervening, or whether these works know which concepts the reader has influenced. This has a meaning only for one individual reader reading the complete set of the locked works: the work’s dialogue is perpetuated in a progression, a progression with temporalities that are disconnected for the author and the reader but which, for each of them, corresponds to an oriented time axis, with no possibility of going backwards. This is a supplementary manifestation of the timeless character (apart from being spaceless) of the functional model as well as of the independence of the author’s and reader’s fields.

The word “locked” is not to be understood as opposed to “open.” It is like phase locking in electronics, the blocking of a process on a particular, imposed functioning. As a matter of fact, the data induced by the reader in a reading will direct the possibilities of generating texts-to-be-seen in the implementations of any textual process. This locking is independent from a chaotic or non-chaotic temporal performance of the work during the readings of the targeted reader, a performance that the future will unveil for us.

Into these locked works each of the subjects, author and reader, will embark on a one-way journey: The author, having to manage in each work what he or she has not yet conceived, cannot erase what he or she has already conceived. In the same way, the reader cannot erase the marks of his or her former readings—the marks enable him or her to read further into the locked works but prohibit his or her reading what a neighbor might read.

In the locked works the work is submitted to an expansion from the author’s point of view: the idea of an individual work does not hold value any more, it only exists in connection with the whole, including the other works; it only achieves its meaning within the body of locked works.

In the locked works, however, the works are subjected to a contraction, from the reader’s point of view. While getting rid of the barriers between its parts, the work behaves, on the whole, like a huge hypertext that the reader sees him- or herself going through. It is the whole body of work that, for him or her, is reduced to a unique work, with possibly multiple aspects—a work that becomes more and more difficult to share over time, since it becomes linked to so many reading particularities: a work that, a little more at each reading, grows while digging its roots into privacy, being more and more distant to other readers.

I hope, dear reader, that having reached the end of this journey you will be convinced—that the actions of these funny authors on their funny machines can result in detailed and coherent writings, not always as trivial as the waving away that they sometimes incite might suggest; that these actions diverge now into completed, in-progress or future productions that correspond to new literary offers and not to an abolition of the forms, habits and customs of earlier literature, but to an enrichment of literature; that procedural computer literature is no longer in its beginnings but has reached maturity.

And that such work is wholly rooted in human being and not in the machine!

Acknowledgment

The author wishes to thank Les Treilles Foundation for providing the translation of this article from the original French.

References and Notes


7. A more detailed description of the model is given in Philippe Bootz, “Un modèle fonctionnel des textes procéduraux” (A Functional Model for Procedure Texts), Les Cahiers du CIRCAV, No. 8 (Villeneuve d’Ascq, France: Rennes, January 1996) pp. 191–216, even though inferred data are not mentioned there. The chart shown in Fig. 1 is

Bootz, The Functional Point of View

315
taken from this article, but a few changes, which were necessary to describe the present state of the approach, have been added here.

8. The realizer is defined as anyone who contributes to materializing the author’s idea. This may often be the programmer or may also be a technician who brings about the realization of some aspect of the project, such as sound or video.

9. Oulipo (OUVRIOIR DE LITTERATURE POTENTIELLE) is a literary group created in France by Raymond Queneau and François le Lionnais in 1961 from which was created A.L.A.M.O. Oulipo developed fixed structures for literature from mathematical rules and made particular constraints on literature. Oulipo worked with constraints at a structural level. In my model, a different functional pattern, if possible, would generate constraints at the functional level that overlay the structural level (in a programmed work, the structure of the text-to-be-seen appears only if the program is run, allowing the functional level to act). One can imagine different functional patterns that are compatible with the actual techniques and also differ from the one presented here. For instance, some works on the Web, with multiple authors, have another functional pattern. The systematic literary exploration of the different possible functional patterns would constitute a meta oulipo.

10. The position opposite the one we are starting with prevails from the reader’s point of view and is also true. We shall come back to it later, but here I would like to give Jean Pierre Balpe’s words for it: taken from Balpe [6] “Méta-auteur,” p. 97; “He or she [the meta-author] can only be placed at a distance from the terminal text [text-to-be-seen in our present terminology], of which he or she does not master all the components.”

11. See Bootz [7].

12. One can also refer to this property as “independent” of the reader’s and author’s fields: the textual objects that are to be apprehended by the author are not the same as those to be apprehended by the reader. Or we could speak of an independence of the author’s and reader’s points of view: the working of the textual process is not the same for the author as it is for the reader. When speaking of “text,” the author and the reader are not always referring to the same thing. The communication process, the work, then becomes part of the tower of Babel, at least concerning the statements and opinions that are to be apprehended by the author and the reader, respectively. The textual process produces texts-to-be-seen with performances identical to those intended by the author. Taking a close look at the generation function shows that these two characteristics are actually incompatible: in order to be readable in different situations, a work must be able to function in all these situations, and it is very unlikely that the functioning will be similar in all situations, a hypothesis that is confirmed by experience. The reverse point of view obviously exists, and both have been mentioned in above since the debate between Tibor Papp and myself in above 3, which brought this problem to the fore. In the model, the position that gives greater importance to readability than to faithfulness is called “functional,” and the reverse position, “spectacular,” (in other words, the text-to-be-seen claims to be a description, aimed at the reader, of the written text). When the faithfulness of the textual process is not ensured, the text-to-be-seen cannot make up a complete description of the written text, even without automatic generation or the intervention of chance or interactivity. This problem does not exist in a non-procedural work, the question of faithfulness being postponed, possibly until the publishing stage, not until the reading stage.

13. One can also refer to this property as “independent” of the reader’s and author’s fields: the textual objects that are to be apprehended by the author are not the same as those to be apprehended by the reader. Or we could speak of an independence of the author’s and reader’s points of view: the working of the textual process is not the same for the author as it is for the reader. When speaking of “text,” the author and the reader are not always referring to the same thing. The communication process, the work, then becomes part of the tower of Babel, at least concerning the statements and opinions that are to be apprehended by the author and the reader, respectively. The textual process produces texts-to-be-seen with performances identical to those intended by the author. Taking a close look at the generation function shows that these two characteristics are actually incompatible: in order to be readable in different situations, a work must be able to function in all these situations, and it is very unlikely that the functioning will be similar in all situations, a hypothesis that is confirmed by experience. The reverse point of view obviously exists, and both have been mentioned in above since the debate between Tibor Papp and myself in above 3, which brought this problem to the fore. In the model, the position that gives greater importance to readability than to faithfulness is called “functional,” and the reverse position, “spectacular,” (in other words, the text-to-be-seen claims to be a description, aimed at the reader, of the written text). When the faithfulness of the textual process is not ensured, the text-to-be-seen cannot make up a complete description of the written text, even without automatic generation or the intervention of chance or interactivity. This problem does not exist in a non-procedural work, the question of faithfulness being postponed, possibly until the publishing stage, not until the reading stage.

14. Note that the functional interactivity, shown by arrows and from the reading function in Fig. 1, is not linked to a simultaneous working (or the working in the same place) of the reading and generation functions, precisely because this interactivity is functional. As a rule, the properties that are put forward in the model are independent of the spatiotemporal performance of the functions and objects it is composed of, which makes it possible to have different spatiotemporal performances in the reader’s and author’s fields.

15. In the present terminology, a computer production is called interactive only if this interaction reaches the reader’s level of consciousness—in other words, if he is an actor to the reading. In reality, some characteristics of functional interaction do not reach the reader’s level of consciousness.

16. See Bootz [1].

17. Sandrine Habera Gouillard, a reading report in modern literature, Artois University, Arras, France (MOTS-VOIR, 1990), an essay for a master’s degree, published by MOTS-VOIR, Ascq, France, 1995. It is evident in this reader’s analysis that the program broke down before its end. This breakdown occurred without generating an error message.

18. The Time of lire, the ISEA multimedia gallery, ISEA, Paris 1995.

19. All terms are related to the functional pattern.

20. It will intervene in the functional model as one of the elements of the “language,” a material that appears in the detailed writing function (Fig. 5). The adaptive generator’s protocol is not part of the author-texts, but the necessary choices relative to its application within the framework of a particular work are.

21. This point of view is not incompatible with the model presented previously, where the text-to-be-seen appears as an object, but in the reader’s field. The functional point of view, however, which is used to finalize the adaptive generator, is an author’s point of view about an object he or she cannot have access to. The way the author perceives it cannot alter the reader’s perception. The adaptive generator uses, in a creative and relevant way, the separation (or disconnection) of the two fields. It is also a functional point of view, independent from the existence or non-existence of a structure that uses it.


24. This fundamental temporality is linked to the unbreakable character of the textual process in operation and not to the presence or absence of animation in the text-to-be-seen. As was seen before, a process exists only if it is in operation. It therefore appears as engraved in time to the person who implements it, i.e., the reader. Breaking a process approaches giving body to the text-to-be-seen, making it nonprocedural and reclassifying it into the category of classical “text objects,” as we have mentioned. It is also altering the status of the object (which generated it into an “artificial author,” a concept that is completely alien to procedural literature.


26. We see materializing, in computer poetry, a very classic situation in electronic arts. The spectator of an electronic art installation plays two different “spectator” roles: the first as the actor of the installation. He or she then is the spectator of something like an “inside” of the work. He or she at that moment becomes the work’s target (the reader’s role contemplated in the functional model is the equivalent of this role of spectator), this state might be called “private spectator.” But the same person is also a spectator of the “scene” of the other spectators playing the work. That state might be called “public spectator” (the functional model shown only contemplates private communication between two individuals and therefore does not include the equivalent “public” reader. The model’s critic cannot play the role, since his or her aim is to look for information, while it is not the spectator’s, who is “witnessing the working out of a work.” For the “public spectator,” the “private spectator” activating (generating) the scene is part of the work’s mechanism as an instrument. This is exactly the role that is contemplated here, but for the same person at different times and not for different persons at the same time. Note that a given person, in an electronic art exhibition, still finds himself or herself placed in both states at different turns. This functioning is retained in computer literature and is the manifestation of a characteristic of procedural works that I have not mentioned yet and which is more fully discussed in other articles: the reading achieves an installation.

27. For the exact meaning of the terms in Fig. 3, see Bootz [7].