Rencontre : an experimental tool for electronic literature

Jean-Hugues Réty - Paragraphe, Paris 8 University
Serge Bouchardon - University of Technology of Compiègne
Jean Clément - Paragraphe, Paris 8 University
Nicolas Szilas - TECFA, FPSE, University of Geneva
Caroline Angé - GRESEC, University of Grenoble 3

INTRODUCTION

For several years, the Paragraph Laboratory, University of Paris 8, has explored new avenues in the field of digital art and literature. In that context, a project is currently ongoing in this lab, in collaboration with the University of Technology of Compiegne and the University of Geneva, supported by the Maison des Sciences de l'Homme Paris Nord. The goal of this project is to design a computer tool for the writing of nonlinear fictions for interactive media and to investigate its impact on both the writing and reading processes.

CONTEXT[1]

Authoring interactive narratives that both provide the user with a significant amount of non-deterministic navigation choices and satisfactory user navigation experience is a difficult challenge. Many researchers have tackled this question during the past twenty years with different approaches and within different research domains.

In the domain of virtual storytelling, agents with human capabilities (reasoning, emotions, verbal and non verbal communication) interact within a fictional world. These fictional worlds most often involve mechanisms for plot structuring based on artificial intelligence techniques like for instance planning algorithms. In this context, stories and narratives are represented into the computer system with computing models for interactive narrative. These computing models often rely on an analytical approach: narrative elements like characters, places, characters actions, inter-characters relationships, etc. must be precisely defined in order to author the narrative. Several such computing models were proposed in the literature [3, 6, 8, 17, 21, 24] (see [7] for a review). They typically define partial orders on narrative events and take into account structuring elements like preconditions, assertions, causal constraints and temporal constraints, as well as narrative constraints like for instance narrative tension [15] or Suspense [20]. Some of these computing models are based on Aristotle [1], Vladimir Propp [16] or Claude Brémond's [5] literature and theatre studies (for instance the IDtension project [23] or research by Michael Mateas [14]). All these computing models differ from their degree of abstraction and from their granularity. For instance, beats necessitate from the author to specify high-level narrative units, reducing the scope of the user's agency [15]. On the opposite, the IDtension system can generate low-level narrative actions.

Authoring stories in this context is a long and difficult process because it requires the detailed specification of a large amount of basic narrative elements (properties of objects, properties of characters…). Authoring interactive narratives requires taking care of the numerous constraints entailed by these narrative elements, as well as taking care of narrative constraints. In order to help authors in this complex process, some research projects are developing authoring environments with integrated sets of authoring tools aimed at helping the author to control all these narrative elements and all these constraints [9].

We discussed in [22] some of the problems faced by interactive narrative authoring. In our opinion, a key issue is that the analytical computing models mentioned above are not natural devices to authors. Authors primarily reason with linear plots and subplots that do not match easily into the models defined by interactive narrative research.

---

[1] The context and the presentation of the tool are already dealt with in a paper for the DIMEA 2008 conference.

Electronic literature is another domain of interest with respect to interactive narrative. Since the late eighties, authors have been investigating the authoring of electronic literature with software like Dreamweaver or Storyspace [4]. Some interesting results were obtained (let us cite the hypertext novel *Afternoon a story*, by Michael Joyce [11]), but electronic literature has failed to produce a large variety of satisfactory narratives. In our opinion, a major reason for this situation is that the hypertext model Dreamweaver and Storyspace are based on hardly allows for the controlling of narrative constraints. Several proposals were made in order to extend the hypertext model, like for instance preconditions on links [4, 13]. This resulted in some improvements, but it didn't solve the more general problem of authoring interactive narratives with hypertexts.

Beside the domain of electronic literature, authoring tools for branching narratives have been developed [2, 18, 19, 10]. Some of these tools provide graphical authoring interfaces based on 2D graph representation. These attempts failed in offering solutions to manage large-scale hypernarratives because as the graph gets bigger, the author hardly maintains a global sense of the overall branching structure.

We think there is a need for interactive narrative authoring tools and engines that do not primarily involve a complex analytical process and that in the same time allow for much more control over the narrative than hypertext-based models and branching narratives. We propose such tools in this paper. Our starting point is the very important role played by structural division in authoring (non interactive) textual narratives. The organization of text into words, sentences, paragraphs, chapters, volumes, etc. helps the author to structure his/her writing. This organization helps the author to maintain a higher-level view on what he/she is authoring, especially when the text grows long. Whenever considering the text from the bottom of this "structural division hierarchy", the author can focus on writing words and sentences. Whenever considering the text from some higher level, the author can focus on ordering paragraphs, sets of paragraphs, chapters… that is on a higher-level structural organization of the narrative. We propose a generalized notion of section, that we name *hypersection*, in order to provide authors with high level structuring elements dedicated to interactive narrative authoring.

RENCONTRE : A TOOL BASED ON HYPERSECTIONS

Rencontre is an experimental tool based on a dynamic link engine, managed by structural entities named hypersections. A hypersection is a recursive container : it can include not only framents, to be directly accessed by the reader, but also hypersections. Simple rules for sequencing/interweaving hierarchical hypersections allow for very variable potential reading trajectories, while limiting the complexity of writing that is observed in classical hypertextual approaches.

This section describes hypersections. In our framework, an interactive narrative is a set of fragments structured with hypersections.

Let us call *fragment* an atomic content element in our framework. Fragment contents are intended to be displayed to the audience. Fragments can contain simple text but can also contain other medias.

Termination of a fragment. Every fragment is associated a termination property. A straightforward example of a termination property associated with a fragment is: the fragment is terminated when it has been displayed to the user who is viewing the interactive narrative (this means that the user will not be displayed twice this fragment). The author of an interactive narrative can specify more elaborated termination properties in such a way that the user will be given access to the same fragment several times.

The aim of the computing model we present here is to provide authors with a simple tool for structuring fragments together. Fragments can be composed together within hypersections, in a recursive manner, as follows:

A *hypersection* is composed of:

- a set of subsections;
- a behavior;
- a termination property.

Subsections are fragments or hypersections. The set of subsections in a hypersection may contain fragments only, hypersections only, or a combination of fragments and hypersections. No circularity is

allowed: a hypersection must not contain itself, directly or indirectly. Hypersections are thus organized in a hierarchical way.

Termination of a hypersection. A straightforward example of a termination property associated with a hypersection is: the hypersection is terminated when all its subsections are terminated. The author of an interactive narrative can specify more elaborated termination properties, for instance in such a way that the hypersection is terminated when a certain percentage of the fragments it contains are terminated.

In our framework, a user reads an interactive narrative by successively accessing fragments. At each stage, the set of fragments that can be accessed by a user is specified by the hierarchy of hypersections, behaviors of hypersections and by termination of fragments and hypersections. We call this set: *set of successor fragments.* The hierarchical structure of hypersections and fragments specify all the different paths users can experiment among fragments. The specific path one user is going to experiment is determined by user interactions.

We define two main classes of hypersection behaviors. These two kinds of behaviors were foreshadowed in a previous work with Robert Kendall [12].

Deterministic behaviours. Reading a deterministic hypersection (i.e. a hypersection with deterministic behaviour) proceeds with respect to the order of its sub-sections: whenever one user accesses a deterministic hypersection, he/she accesses its first non-terminated sub-section. To be more precise, the set of successor fragments of a deterministic hypersection is the set of successor fragments of its first non-terminated sub-section.

Deterministic hypersections provide a structural organization where subsections are organized into sequences. This kind of hypersection is particularly dedicated to plot structuring. Plot structures require the satisfaction of causal constraints and narrative constraints. The simplest way for enforcing such constraints is sequencing.

Non-deterministic behaviours. Reading a non-deterministic hypersection proceeds without any predefined order: whenever one user accesses a non-deterministic hypersection, he/she can access any non-terminated sub-section. To be more precise, the set of successor fragments of a non-deterministic hypersection is the union of the sets of successor fragments of its non-terminated subsections.

Non-deterministic hypersections provide a structural organization where no particular order on subsections is specified. This kind of hypersection is particularly dedicated to narrative descriptions: character descriptions, global situation descriptions, descriptions of locations where the action takes place, etc.

These two types of behaviour offer basic tools for structural organization of fragments altogether. The expressiveness of the framework we propose in this paper results from their combination. A deterministic hypersection can contain (directly or indirectly) non-deterministic hypersections, and vice versa.

When authoring first examples of narratives within our framework, we found that deterministic behaviours entailed reading problems. When accessing the last fragments of a subsection, the reader was given a decreasing number of choices. At the end, he only had one choice! We thus introduced a new type of behaviour.

Deterministic behaviours with fuzzy transitions. Reading a deterministic hypersection with fuzzy transitions proceeds with respect to the order of its subsections, except that the reader can be given access to a subsection B before the preceding one, say A, is terminated. A threshold specified by the author defines the moment when the reader can access B. For instance, if this threshold is 70%, this means that whenever one reader has read more than 70% of A, then he/she can both access A and B. The author can also set up a mechanism that insures that the reader will terminate A before going too far in reading B. This is defined with another threshold. For instance, if this threshold is 20%, this means that if the reader reads more than 20% of B, then he/she is forced to terminate A before going further in reading B.

A GRAPHICAL AUTHORING TOOL

The tool is developed in Java with Eclipse. It currently contains some writing and reading features such as the folding/unfolding of the structure, the zooming or the colour coding. The tool is developed in collaboration with authors and readers. This design methodology aims at experimenting several alternative interfaces.

The tool is now composed of various modules: the writing module, the structure visualisation module and the reading module (itself composed of a fragment reading interface and an inter-fragment navigation interface).

The writing module

Authoring an interactive narrative within our framework is writing a set of fragments together with organizing these fragments into a hierarchy of hypersections. Our assumption is twofold.

First, we assume that this simple framework has yet enough expressive power to allow for authoring interesting interactive narratives (we foresee extensions to what is presented in this paper in order to strengthen further this expressive power). A short interactive fiction with 38 fragments was already authored, and we are currently working on a "real size" experiment with a recognized author.

Second, we assume that authoring within our framework allows for a rather good control of the author over the overall work. In other words, we expect our framework to overcome the "lost in hyperspace" syndrome that authors (and readers…) often encounter whenever the size of an interactive narrative grows high. Our answer to this challenge is based on the one hand on the notions of hypersection and behavior that are simple enough to allow for a good understanding by human of a combination of hypersections into a hierarchy of hypersections. On the other hand, the notion of hypersection (and more specifically hierarchical composition of hypersections) allows for efficient visual representation of an interactive narrative structure into a tree of hypersections.

We designed a visual representation of a hierarchy of hypersections into a 2D tree. This representation was implemented in Java using the JTree library. Figure 1 shows a part of the hypersection hierarchy of the short fiction we wrote with hypersections.
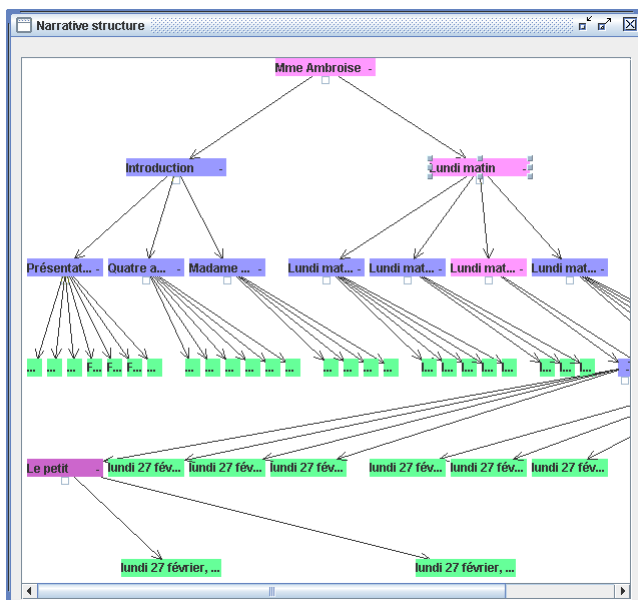

Figure 1. Hierarchy of hypersections represented into a 2D tree. Fragments are in green, non-deterministic hypersections in blue, deterministic hypersections in violet, and fuzzy deterministic hypersections in light violet.

The authoring tool includes a structure editor based on this 2D tree representation. The editor allows for creating, editing and modifying a hierarchy of hypersections. For every node, a rollover frame gives details as shown in figure 2. Let us notice the *nbChoicesMax* parameter that appears in this figure. This parameter can be associated with any hypersection node. It constrains the cardinality of the set of successor fragments to be less or equals to nbChoicesMax. In other words, the reader tool (see

section 4) will propose to the user no more than nbChoicesMax selection choices at each step when accessing the hypersection. This mechanism prevents cognitive overload of the reader with too many alternative choices.
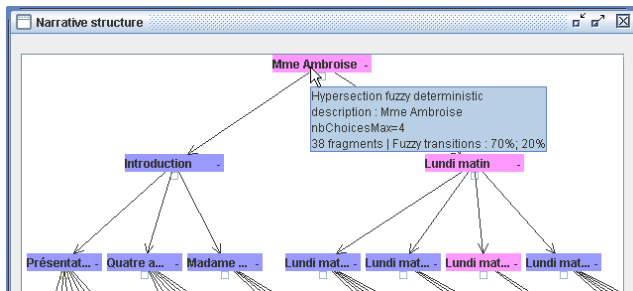

Figure 2. Rollover

Fragments are edited as shown in Figure 3. Fragment contents are coded into HTML. They can be modified with an integrated HTML editor. Hypersections can be edited in similar way. The interface allows for adding, suppressing and moving nodes.
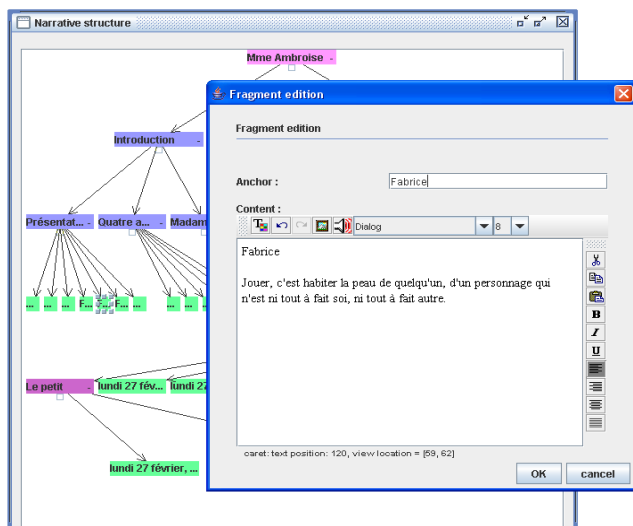

Figure 3. Edition of a fragment

Zooming and folding/unfolding of subtrees are two important features. The author can work on global organization of the narrative with viewing the entire tree or large parts of the tree. He/she can also focus on small parts of the tree with zooming or folding subtrees. Figures 4 and 5 show these functionalities.
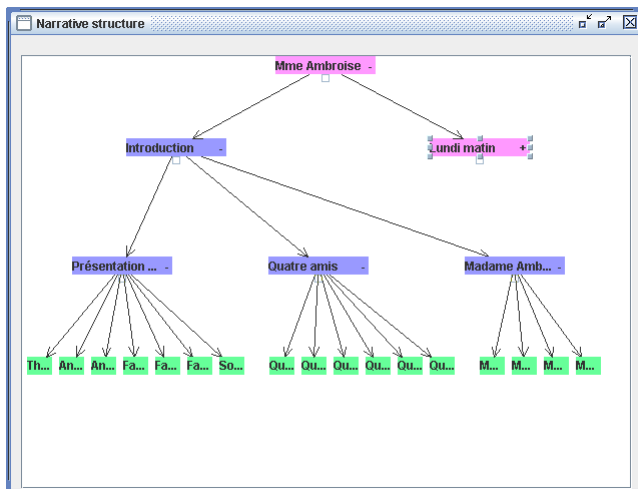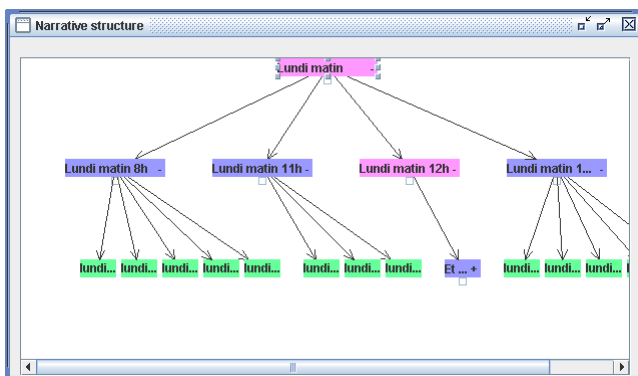
Figure 4. Folding

Figure 5. Zooming

Interactive narratives authored with this authoring tool are stored in an XML format. The authoring tool can import and export this format.

The reading module

Reading an interactive narrative authored with the authoring tool described above requires a reader software. This reader tool imports an XML file and displays fragment contents to the user.

There are many ways reader softwares could manage interactivity with the user and render fragments contents. We describe in this section two reader tools we implemented in Java.

Whenever a user reads a hypersection-based interactive narrative, he/she eventually reads a sequence of fragments. The algorithm that determines what sequence of fragments (what fragments, in what order) a user actually reads is a non-deterministic algorithm. This means that at each execution step (that is: each time a fragment is being selected for being displayed to the user), there is a set of fragments that are candidate for being selected next (this set is the set of successor fragments of the top level hypersection in the hierarchy). Interactivity can occur at this level: the user can be asked to select a fragment among the set of fragments candidates.

A non-interactive reader

Let us consider a simple reading tool where fragment selection is made by the machine on random bases. No user action is required. We then obtain some kind of a narrative generator: each execution of such a reading tool results into some randomly generated particular ordering of fragments that can then be put together into a single linear document. Let us notice that every ordering this generator can produce is obtained with the non-deterministic algorithm mentioned above. This means that every such ordering is consistent with the hierarchy of hypersections, and especially with behaviors of hypersections.

Interactive reading tools let the user, at each reading step, choose among a set of fragments candidates. Such reading tools are thus composed of two parts: a fragment viewing interface and a selection device.

The viewing interface is responsible for displaying fragments. In our current implementation of the hypersection framework, fragment contents are coded in HTML (without links: HTML is here only used to format fragment contents). The viewing interface is thus a simple HTML viewer.

The selection device is responsible for letting the user select the next fragment to be displayed. Many selection interfaces can be imagined. Figure 6 shows the 3D selection devices we implemented in Java3d. The user manipulates (rotates) the 3D sphere with the mouse. He/she then clicks to select an "anchor". The corresponding fragment is displayed in the fragment-viewing interface, and the selection device is refreshed with new anchors (from the new set of fragments candidates).

Let us make more precise the notion of anchor. In an interactive narrative setting, the reader must base interaction on some kind of prediction of what effect his/her selection (and more generally: his/her action) is likely to produce on the narration or on the fictional world. We decided to associate every fragment with what we call an anchor (by analogy with anchors in hypertext) because anchors bear a meaning that helps the reader making selection choices. In our current implementation, anchors, just like fragment contents, are coded in HTML. We expect however in future work developing a hypersection based framework where contents are actions and "anchors" are commands for actions, in such a way that reader softwares could involve virtual characters.
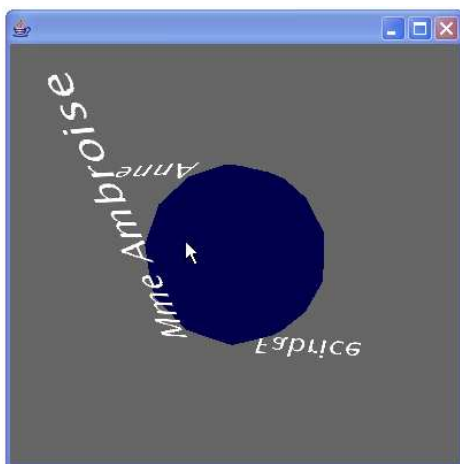

Figure 6. A 3D selection device

The structure visualisation module

The structure visualisation module, initially designed for monitoring and feedback purpose for the author quickly appeared as a relevant module, on the reader's side (cf. next section). In the figure above, this module is used as a reading interface.
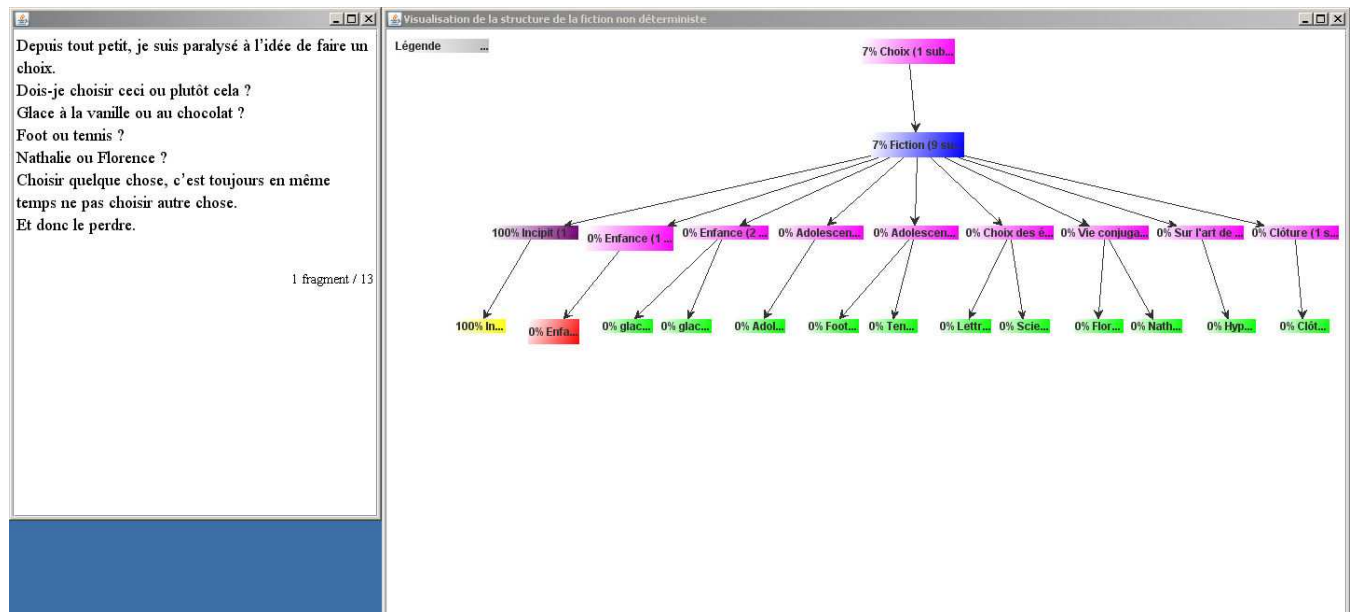
Figure 7. The structure visualisation module.

PERSPECTIVES FOR COMPUTER-BASED LITERARY CREATION

Reading of the text vs reading of the device ?

As previously said, the structure visualisation module appeared as a relevant module for the reader. Indeed, it stages the reading process in a unique and innovative manner. The reader is able to combine the reading of the text and the reading of the device. He/she can enjoy the discovery of sophisticated mechanics for reading management. He/she can also realize that some unread fragments are still to be discovered. More generally he/she realizes that other trajectories are possible and can enjoy the re-reading of the piece.

Does this new approach present a risk of a displacement of the reader's interest in favor of the device, at the expense of the literature quality itself ? We hypothesise that the reading of the device is fully part of the reading pleasure.

A fragmentary literature

Rencontre, as an experimental tool for electronic literature, is questioning the possibility of a nonlinear digital literature. Many researchers have already pointed out the links between the hypersection-based literature and the fragment. According to them, Hypertext would be a concrete expression of broken down texts. This analogical connection araises other questions and we can agree with Jean Clement to say that « Hypertext is halfway between the fragment and the linear text ». Although, *Rencontre* is not a simple hypertext. As an interactive writing tool for non-deterministic narratives, it changes the relation between the digital text and the fragment in many ways.

If we consider a fragment from a scriptural perspective as « a discursive unit that is independent from the linear writing and reading, but whose independency is not compatible with the piece as a whole »[2], then *Rencontr*e belongs to this genre. From the writing perspective itself, there is a fragmentation of the narrative's fragments. This fragmentation lies in the launch of a non linear narrative carried out by internal functionalities of the tool. The hypersection provides the author with giving access to the reader to fragments to define properties that will define the behaviours of the hypersections.
Therefore this random order oscillating between opening and closing demonstrates the practice of a fragmentary writing. The fragmentary genre belongs to a provoked and organised plurality.

---

[2] Angé Caroline, « La question du sens. Ecrire et lire le fragment. Du texte à l'hypertexte, Thèse de doctorat, Université Paris 13, 05.

Futhermore, the impact of this tool crystallizes the relation to the fragmentary aesthetics. How the fragment appears and how it itself relies on the appearance of "untied" and "caught up", on the part and the whole. The structure visualization device, associated with a reading mode, shifts the fragmentary aesthetics from the text towards the reader. The reflexivity created by this reading mode generates *a negotiated fragmentation*. The selected narrative sequences give the reader a control on the text appearance while building up a new text. The potential manipulations like the unfolding/refolding of the structure –fragmentory reading interface- correspond to this work of elaborating links left to the interpretation of the reader.

We can wonder if the reading supported by this tool leads to a new reading cooperation in the digital fragmentary literature.

At last, from a morphological perspective, the spaciality of the text produced via the 3D interface is a symbol of the visual game of the fragmentary aesthetics. Beyond the symbolical dimension, the fragment is staged as it is. The visual and narrative tool gives to a certain extent a visual existence to the text fragments… The fragment is upgraded as an image.

Chance and meeting : towards a randomness literature

The concept of random is very old, but its introduction into contemporary arts is due to the discovery of its epistemological importance in modern science. If Einstein could still argue that "God does not play dice," the discoveries of quantum physics, fractal, complexity and chaos have changed our world view and have questioned the artists very early.

In the literary field, the first random works can be classified under the paradigm of combinatorics. The process is to offer the reader fragments of text the arrangement of which is indeterminate and which are designed so that any combination produces a reading effect. These effects of reading can range from pure deconstruction of meaning, as in Tzara, for example, to dizziness facing a process able to generate ad libitum readable texts as in Queneau or Perec works.

The arrival of computers in creative writing is contemporary (not a coincidence) with the rediscovery of combinatorics by members of the OULIPO in the sixties. The random selection of text fragments by the computer makes it a powerful machine to read / write combinatorial works. The first programs were content with aligning words or phrases drawn randomly in a textual database. Thereafter the authors have sought to introduce coherence in the statements generated by the machine, using algorithms inspired by generative grammar or textual grammar.

Other works, inspired by cybernetics, have sought to make the reader a partner of the author and of the programme in the establishing of texts given to read on the screen. The most popular form of this interactivity in the field of literature has long been hypertext. In works of literature hypertext, it is no more randomness which is the basics of reading, but rather the serendipity, understood as the process of finding something interesting unexpectedly, seeking something else, or even without knowing what you are looking for. This concept often invoked scientific research finds its equivalent in literature in the notion of chance. The surrealists had made it a cornerstone of their approach to the poetic as "*hasard objectif*," the unexpected realization of an unconscious desire. There is, in reading a hypertext writing, something of this chance encounter, the reader clicking on a link in anticipation of the text to come, but uncertain about what will be given to read.

From a certain point of view, the software *Rencontre* plays with these two distinct but related concepts that are randomness and chance. At first glance it appears as a hypertext, including its 3D version. At each step, the reader is invited to select the continuation of the story by clicking on a link. He/she moves warily forward, drawing a narrative journey in which he/she feels involved. But unlike a hypertext, the story he discovers is the same whatever his/her reading choices. Only the narrative of this story is variable, depending jointly on the random function of the computer that provides links and on the choices of the reader who clicks them. For the author, this is a new challenge: he/she must both abandon the total control of his narrative while controlling the randomness of the machine. The software *Rencontre* provides the author with the means to meet this challenge and presents the reader with the pleasure to discover a variable narrative in which reading turns into playing.

CONCLUSION

Rencontre is an experimental tool for electronic literature. A graphical authoring tool and a reading tool with a simple 3D interaction device were implemented in Java. We authored a short interactive fiction and we are currently working with a professional author on a longer fiction.

This ongoing project will now develop in several directions. Once the new fiction will be finished, two kinds of evaluations will be conducted. From the author point of view, the usability of the authoring tool and the conceptual model of hypersections will be assessed (via interviews), in order to refine the tool and possibly develop new metaphors of authoring with hypersections. This evaluation takes place in an iterative design, in which the authoring tool and the engine are refined while new art pieces or demos are produced. From the reader point of view, the digital narrative will be tested on a panel of readers, in order to evaluate how the readers perceive the hypersection-based underlying model. In particular, we hypothesize that the user might perceive the difference between a simple graph and the hypersection-based model, the main contribution of the model being for the authoring process.

Further research and development is also needed on the reader tools. Indeed, the way the reader can both navigate the fragments and visualize them has a deep impact on the whole reader experience. Our goal is to provide a panel of different tools, customizable by the author, in order to enhance author's expressive power within the proposed framework.

The proposed tool opens new perspectives to computer-based litterary creation, since it opens a certain form of procedural writing to non computer expert authors. In particular, the visualisation of the structure for the reader provides the author with the opportunity to invent new discursive strategies.

ACKNOWLEDGMENTS

REFERENCES

[1] Aristotle, 330 BC. The Poetics. Mineola, New York: Dover, 1997.

[2] Balet O., INSCAPE: An authoring platform for interactive stories, in "Virtual Storytelling. Using Virtual Reality Technologies for Storytelling" (Marc Cavazza, Stéphane Donikian Eds.), Lecture Notes in Computer Science 4871, Springer 2007

[3] Bates, J., Virtual Reality, Art, and Entertainment. In Presence: The Journal of Teleoperators and Virtual Environments, -1(1), MIT Press, Winter 1992.

[4] Bernstein M. Storyspace 1. In Proceedings of ACM Conference on Hypertext and Hyper-media, College Park, Maryland, 2002.

[5] Bremond, C. 1974. Logique du récit. Paris: Seuil.

[6] Cavazza, M., Lugrin J-L., Pizzi, D., and Charles, F., 2007. Madame Bovary on the Holodeck: Immersive Interactive Storytelling. ACM Multimedia 2007, Augsburg, Germany.

[7] Cavazza, M. and Pizzi, D., Narratology for Interactive Storytelling: a Critical Introduction. 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, Germany, December 2006.

[8] Cavazza M., Charles F., Mead S.J., Character-Based Interactive Storytelling. In IEEE Intel-ligent Systems 17(4): 17-24, 2002

[9] Donikian, S., Portugal, J.-N. Writing Interactive Fiction Scenarii with DraMachina, in Pro-ceedings of TIDSE'04, 2nd International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, 2004.

[10] Gordon A. S., Authoring Branching Storylines for Training Applications. Proceedings of the 6th international conference on Learning sciences, 2004.

[11] Joyce M., Afternoon, A story, Ed. Eastgate Systems, Watertown (Massachusetts), 1990.

[12] Kendall, R., Réty, J.-H. Toward Organic Hypertext. In Proceedings of ACM Conference on Hypertext and Hypermedia, San Antonio, 2000.

[13] Kendall, R., Réty, J.-H. Connection Muse. Web-based hypertext poetry and fiction authoring system. Developed from 1999 to 2001. http://www.wordcirtuits.com/connect

[14] Mateas M. A preliminary poetics for interactive drama and games. In proceedings of SIGGRAPH, 2001.

[15] Mateas M., Stern A. Structuring Content in the Façade Interactive Drama Architecture. In Proceedings of Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2005), Marina del Rey, 2005.

[16] Propp, V. 1928. Morphologie du conte. Ed. Seuil, Paris.

[17] Riedl, M. O., Young, R. M. An Intent-Driven Planner for Multi-Agent Story Generation. Proc. of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 186-193, 2004.

[18] Silva A., Raimundo G., Paiva A., Tell Me That Bit Again... Bringing Interactivity to a Virtual Storyteller. International Conference on Virtual Storytelling 2003: 146-154

[19] Silverman, B. G., Johns M., Weaver R., Mosley J., Authoring Edutainment Stories for Online Players (AESOP): Introducing Gameplay into Interactive Dramas, In proceedings of International Conference on Virtual Storytelling(ICVS'03), novembre 2003.

[20] Somanchi, S. K. 2003. A Computational Model of Suspense in Virtual Worlds. Technical Report Number 03-002, Liquid Narrative Group, North Carolina State University.

[21] Szilas, N. 2007. A Computational Model of an Intelligent Narrator for Interactive Narratives. Applied Artificial Intelligence, 21(8), 753-801.

[22] Szilas N., Marty O., Réty J.-H. Authoring Highly Generative Interactive Drama. In Pro-ceedings of International Conference on Virtual Storytelling, Toulouse, 2003.

[23] Szilas N. IDtension: a narrative engine for Interactive Drama. In proceedings of 1st Inter-national Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE), 2003.

[24] Young, R. M., Riedl, M. O., Branly, M., Jhala, A., Martin, R. J., & Saretto, C. J. 2004. An architecture for integrating plan-based behavior generation with interactive game environments. Journal of Game Development, 1(1), 51-70.