The Shattered Document Approach to Adaptive Hypertext. Design and Evaluation

Mário AMADO ALVES [a,1]
[a] *LIAAD-INESC TEC.* amado.alves@gmail.com

Abstract. We address the problem of improving, automatically, the usability of a large online document. We propose an adaptive hypertext approach, based on splitting the document into components smaller than the page or screen, called *noogramicles*, and creating each page as a new assemblage of noogramicles each time it is accessed. The adaptation comes from learning the navigation patterns of the *usors* (authors and readers), and is manifested in the assemblage of pages. We test this model across a number of configurations, including chance and non-adaptive systems. We evaluate our model through simulation. We have designed a simulator based on established findings about the behaviour of hypertext users. We have realised a quantitative evaluation based on hypertext usability measures adapted to the problem: session size, session cost.

Keywords. H.3.3 Information Search and Retrieval / Information filtering, Retrieval models, Search process, H.5.1 Multimedia Information Systems / Evaluation/methodology, H.5.4 Hypertext/Hypermedia / Theory, I.2.0 General / Cognitive simulation, I.2.6 Learning / Connectionism and neural nets, Algorithms, Design, Experimentation, Theory, adaptive hypertext, spreading activation

1. Introduction

We study the large online document, and how its utilization might be improved by means of adaptive hypertext features. *Large* means an extent such that the document cannot be seen all at once. In other words: large = (much) larger than a screenful.

The overall process of hypertextualization and hypertext adaptation is depicted in figure 1. The *Document* state represents a conventional document, or else a poorly structured para-document like a set of forum posts, a wiki, etc. The *hypertextualization* step consists in casting this content into an *Hypertext*ual form that can be adapted. The aim of *adaptation* is to have this form evolve onto a *Better hypertext*, by learning from the navigation patterns of the users (or some other adaptative input).

A classical example of adaptive hypertext at the service of improved utilization of a large online document is the *Knowledge Sea II* system [1], applied to a manual of C, in a programming language learning environment.

[1]This work is financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP - 01-0124-FEDER-022701.

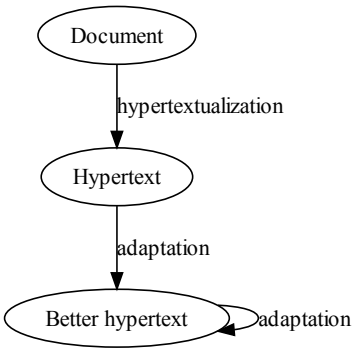


Figure 1. Adaptive hypertext: overall process

In our work, the hypertextualization step entails splitting—*shattering*—the Document into its atomic constituent parts, which we call *noogramicles* (small representations of knowledge). We detail this model later. We use as adaptative input the choices of the users. The adaptative output consists in newly created pages, assembled from the most connected noogramicles; therefore we call this process *renoding*—paraphrasing the classical method of relinking [5].

1.1. Operational definition of hypertext

There are various kinds of hypertext, cf. [9], [7] and references therein. A consensual definition is missing at large. From the examination of the various kinds of hypertext, we have found the following invariants, the set of which constitutes our operational definition:

- hypertext is an interface to interconnected items
- the items are of information, textual or pictorial
- the interface lets the user follow any connection
- the interface records the connections followed, and lets the user relive them at will; in particular, the interface provides a back button

1.2. Article organization

On section 2 we describe the shattered document approach, which comprises a document model (section 2.1) and an adaptation model (section 2.2). On section 3 we describe our experimental design, namely the simulator (section 3.1) and the quality measures used to evaluate the configurations (section 3.2). On section 4 we describe a selection of the configurations experimented with and their results. We conclude on section 5.

For space reasons, in this article we had to leave out a number of items, notably the rationalia for hypertextualization and renoding, the description of the method of spreading activation, and the description and results of a large number of experimental configurations pertaining to random users and alternative document strutures. Such items are fully described elsewhere [3,2].

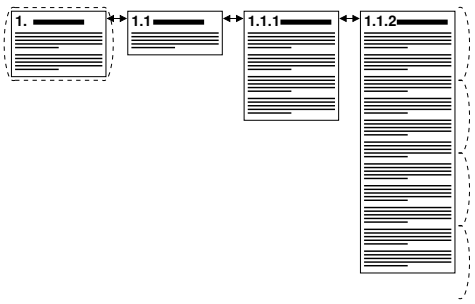


Figure 2. Standard hypertextualization of the sequential structure.

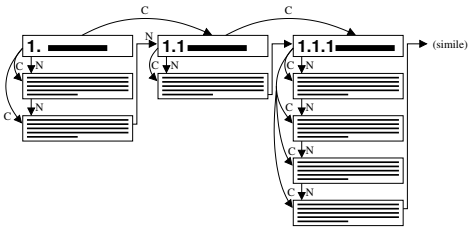


Figure 3. Model of the same document in figure 2 but with the shattered document approach and the two types of connection Next (N) and Child (C).

2. The Shattered Document Model

2.1. Document Model

The Shattered Documents approach prescribes that documents be taken apart into their smallest constituents of meaning, or *noogramicles.* Naturally the noogramicles must connect with each other, in order to create, ultimately, a navigationable network, or hypertext.

We look mostly at hypertext documents created from traditional documents, for example the hypertext version of the *Ada Reference Manual* or ARM [4], which we use for experimentation. We have observed three dimensions in the traditional structure of documents—sequence, hierarchy, cross-reference. We transcribe these dimensions into types of connection in the network.

Figure 2 depicts the standard hypertextual edition of the ARM. The representation, albeit stylised, honors the actual data, in the numbering and size of the sections, and the links thereof. Each section is a node, or page—an integral HTML file. The pages are connected by *Next* and *Previous* links. Four sequent nodes in this sequence are represented. The links are symbolized by the arrows in figure 2, and designed on the interface as buttons located at the top and at the bottom of each page.

Figure 3 shows how the same part of the ARM in 2 is modelled with the shattered document approach and the two types of connection Next and Child. Figure 4 shows a page made up of the first five noogramicles in the model; the constituent noogramicles of this page are selected using spreading activation from the first one in a manner detailed later.

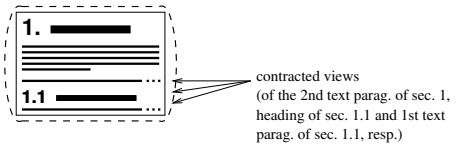


Figure 4. Page made up of document fragments.

To consolidate: a document is represented as a graph, or network datamodel of noogramicles interrelated by directed connections of three types—*Next, Child, Refer*—, as follows.

Next represents the linear order of paragraphs. Example: from a paragraph to its immediate successor. Note that, by the extended paragraph definition, Next also connects from a section heading to the first classic paragraph of the section, and from the last classic paragraph of a section to the heading of the next section.

Child represents the immediate subordinate relationship between paragraphs. Examples: from a section to each of its subsections; from a paragraph introducing an enumeration (e.g. a bulleted list) to each item of the enumeration; possibly, even from a terminal section (i.e. a section without subsections) to each of its paragraphs

Refer represents other reference relationships. Examples: from a paragraph to a footnote; from a paragraph to another paragraph or section (e.g. the so-called *cross references*, and *see also* references); from a index entry to its target paragraph or section; from a TOC entry to the corresponding section.

In the Child and Refer relationships, a section is represented by its first paragraph, normally a heading. Original references anchored in sub-paragraph units (e.g. words) are represented as references anchored on the paragraph as a whole.

As we are targeting an *adaptive* system, we need a way to represent the corresponding information. The adaptative part of our model is compound of two main items:

Pages. What the reader sees. A page is assembled from a small number of noogramicles, in a manner detailed later. Naturally the user may navigate to another page. Pages are the *adaptive output* of shattered documents.

Travels. The navigation steps that usors (authors and readers) make in the document. Each travel is recorded, and used in adapting the construction of pages. Travels are the *adaptive input* of shattered documents.

The *adaptative process* integrates the two items, by assembling pages based on travel information. The main idea is to select the noogramicles that are most connected to the current one.

So, we must add the connection type Travel to the trio Next, Child, Refer already explained. Therefore, so far our document model is a network of noogramicles with four types of connection: Child, Next, Refer, Travel.

In the current experimental configurations, we interpret each of Child, Next, Refer, as Travel. That is, we unify all types into one. This reinterpretation of the traditional document structure connection types Child, Next, Refer allows us to solve the *cold start problem*, and simplifies immenselly the process of spreading activation. This magic step is justified mainly because, if you look at it, the connections Child, Next, Refer are indeed

the *travels* that the author intended the reader to make in the first place. *Next* is directly so. *Child, Refer* are *contigently* so—they are the paths laid out by the author for the reader to cross, *wanting*. Or, *Child, Refer* carry a *rhetorical* value—which amounts to the same effect (a contigent choice by the reader to follow).

So, a *shattered* document is represented as a graph, or network datamodel of noogramicles interrelated by directed connections representing travels: the paths either actually taken by users or else suggested by the author.

2.2. Adaptation Model

Our main architectural ideas are as follow. The system is an interface into a large document. The interface unit is the page. Pages are accessed one at a time, normally. The document as a whole is partitioned, shattered, into fragments smaller than the page, called *noogramicles*. Each page is an assemblage of noogramicles.

Each noogramicle has two renderings, or views: expanded, contracted. The expanded view is the noogramicle itself, normally. The contracted view is a clear label of the noogramicle. Occasionally, the label equates the noogramicle, i.e. their contracted and expanded views are formally identical.

Figure 4 exemplifies one page in our design. The noogramicle on the top is *central*, and represents the page for certain purposes, explained below. On this design, the real estate on a page is divided aproximatelly equally between expanded and contracted noogramicles. The higher-ranking noogramicles are expanded.

Navigation, or *travelling*, on this design, is effected by *recentring* on a noogramicle, normally by clicking on it.

The main *adaptive input* of our system consists of the choices, or travels, made by the usors. These travels are memorized in the computer as connections between the respective noogramicles. The target point of the connection is the noogramicle clicked on. The source point of the connection is the central noogramicale of the current page, normally. This configuration is called *central-to-central*. Other configurations are possible and were tested, e.g. *all-to-central*, but are not reported in this article.

This graph model is then explored (in the computer) by means of spreading activation [3], to find the noogramicles more related to the central one (which has been chosen by the usor). Such noogramicles then form the new page—the *adaptive output*. Namely, energy is spread from the central noogramicle, and the n most active noogramicles are selected for the new page, where $n = 10$ in the configurations using this method (exposed later).

3. Experimental design

3.1. The Simulator

We have constructed a simulator of user utilization of an adaptive hypertext system, to experiment and compare different configurations of the adaptive system, automatically. The design bases of the simulator include the concept of *oracle*, and the *Smart User Assumption*.

The simulator creates sessions of utilization of a hypertext. The simulator comprises two components: the Oracle Model and the User Model. Each session is a user's quest for

the noogramicle that will respond to their reference question, or information need. Such noogramicle is called the *oracle* of the session. The simulator draws a random oracle for each session. The random distribution of oracles is called the *Oracle Model*.

The simulated user, or *User Model* consists in a function Choice which selects a page item. It acts as the user clicking on a hyperlink to follow. The *Smart User Assumption* asserts that a human user chooses the right link if the link label is clear about the distal content. For the case when the oracle is only one click away, the label clearly identifies the oracle, and therefore the user selects the item easily. When the oracle is further away than one click, the intelligence or intuition of the user takes place to select the item most likely to lead to the goal. The results in [5] and [10] are interpretable as support for this assumption.

Given these premises, the Choice function can be designed as a choice of the page item most likely to lead to the oracle. In our implementation this is done via spreading activation from the atom, until a page item X is reached, such that X has not been seen or visited before in the same session. The most active item represents the item most connected to the oracle, globally.

The session ends upon reaching the oracle (successful session), or else a maximum session size (number of pages) has been breached (unsuccessful session).

3.2. Quality measures

We introduce the hypertext usability or quality measure of *session cost* as a refinement of the well established measure of *session size* = number of pages = number of clicks - 1. Session cost is a combination of session size with a few extra factors of *cognitive effort* associated with poorer or null navigational aids, longer pages (requiring scroll), and a large quantity of links (requiring more examination). This is necessary because we want to compare across different configurations, and, on the limit, a configuration of only one large page containing the entire document has always the best possible session size of 1—thus unfairly winning the competition even before the start!

It sounds natural, practical, reasonable, to extend the unit of session size, the page, to this new unit of session cost. Let us call it the session cost point, or just point. We formalize session cost as the sum of the cognitive effort terms involved. We have identified the extra cognitive factors described below. For each factor, we introduce a formula for its contribution to the cognitive cost. These factors affect the page. Naturally, the session cost is the summation of the page costs.

We quantify cognitive effort in points directly related to the session size component. We establish a basis: for a configuration of no scroll and a fixed, low page size (quantity of page items, or links) of at most 10, we equate session cost with session size, and therefore define a fixed page cost value of 1 point. From this basis we define page cost for non-trivial pages as follows.

Scroll cost. The cognitive cost associated with pages longer than the screen is related to that length [11].

$$\text{Scroll\_Cost} = \lfloor p/s \rfloor \tag{1}$$

where p is the page length, and s the screen size, measured in the same units, e.g. characters.

Table 1. Principal statistics compared

Configuration	Success	Size_0	Size_1	S-Gain	Cost_0	Cost_1	C-Gain
Shattered Document	0.94	5.62	3.48	★*1.62*	5.62	3.48	★*1.62*
Shattered Random Document	0.98	4.10	★*3.40*	1.20	*4.00*	★*3.38*	1.18
Structural	1.00	*3.48*	3.48	1.00	*15.20*	*15.20*	1.00
Markov Chains	0.84	5.84	4.22	1.38	5.84	4.22	1.38
Shattered Document with Random Clicks	0.02	*6.90*	*6.98*	*0.99*	6.90	6.98	*0.99*
Total	3.77	25.94	21.56	6.20	37.56	33.26	6.17
Average	0.75	5.19	4.31	1.24	7.52	6.65	1.23

Choice cost. If the number n of links exceeds the magical number 7 minus or plus 2 (cf. [8]), then an extra cognitive effort is imposed upon the user, proportionate to the number of links. From our basis of a fixed cost of 1 point for 10 or less items (close enough to the magical number of 7 + 2 = 9), we derive:

$$\text{Choice\_Cost} = \max(1, n/10) \tag{2}$$

3.3. Result statistics and configurations

We computed a number of statistics to evaluate the results of size and cost: moving average over the whole sequence of sessions, with a window of 50 sessions; average per oracle.

The configurations under experiment are described together with their results in the next section.

4. Experimental results

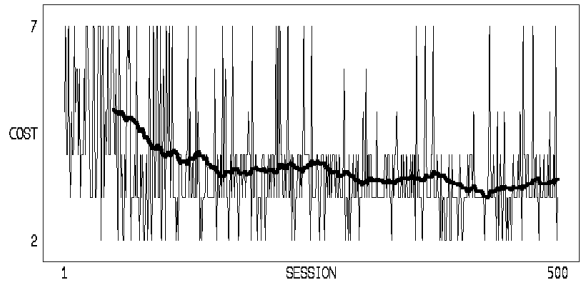
4.1. Overview of results

Table 1 compiles the principal results for each configuration. The *minimum* and *maximum* values in each column except Success_rate are *emphasized*. The *best* final and gain scores in their columns are ★starred. The statistics in the table are a transposition of selected statistics of the involved configurations, mapped as follows:

Success = Micro-average of the success rate of sessions
Size_0 = Average of first session size

Figure 5. Results of configuration Shattered Document

Success_Rate	=	0.94
Size_0	=	5.62
Size_1	=	3.48
Size_gain	=	1.62
Cost_0	=	5.62
Cost_1	=	3.48
Cost_gain	=	1.62



Size_1 = Average of last session size
S-Gain = Micro-average of the gain in size = $\frac{\text{Size}_0}{\text{Size}_1}$
Cost_0 = Average of first session cost
Cost_1 = Average of last session cost
C-Gain = Micro-average of the gain in cost = $\frac{\text{Cost}_0}{\text{Cost}_1}$

We remind that our statistics of *gain* represent a *betterment* of the results, and does not denote numerical increase of the quality measures. In fact, *gain* statistics correspond to a numerical decrease of the quality measures, because, for these measures, less is better. In rigour, as numbers, they are *inverse* quality measures. So, the *gain* statistics invert the numerical relation to present a positive value, i.e. a *more is better* value.

In general, we locate the most important statistics to the right of the tables. And to the bottom in individual configuration tables. Ultimately, we look at Cost_gain. A configuration with greater Cost_gain—all other things being equal—is the winner. But, of course, it is never the case that all other things are equal, hence the need to have the remaining data reported and analysed as well, for a correct interpretation of the results.

4.2. Shattered Document

See figure 5 for the quantitative results of this configuration, and a graphical depiction of the evolution of session cost thereof.

This configuration is our shattered document design of adaptive hypertext, with the original structure of the legacy document, namely a version of the ARM cut down to 1000 noogramicles. This 1000-node structure is a tree of five levels (including root), with an average fanout of 17.85, and an average distance from the root of 3.46—which corresponds to an expected average session size, in the non-adapted version, of 4.46

We observe that the evolution is positive, with a final size lower than that expected value of 4.46 for the non-adapted version. This result shows that adaptation does improve the utilization of a hypertext document.

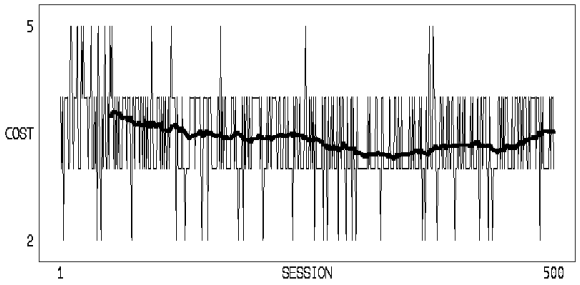
We observe that the evolution happens quickly, in the first circa 200 sessions.

Note that the session cost equals the session size in this configuration, because there is no extra cognitive effort (cf. above). There is no scroll (each page fits in one screen) and the number of choices per page is fixed at ten (the magical number, cf. above).

We observe that the starting size is greater than the expected average for the non-adapted document. Probably this is due to the small size of the shattered document pages

Figure 6. Results of configuration Shattered Random Document

Success_Rate	=	0.98
$Size_0$	=	4.10
$Size_1$	=	3.40
Size_gain	=	1.20
$Cost_0$	=	4.00
$Cost_1$	=	3.38
Cost_gain	=	1.18



compared to the original. Whereas the original pages each contain links to all its descendents nodes, a shattered document page has a fixed number of total items of 10, which is less that the average fanout of 17.85 of the original; therefore, there are oracles that, in the original document are just one node away on a 20-item page (say), but in the shattered document might require an extra, intermediary 10-item page to be visited.

4.3. Shattered Random Document

See figure 6 for the quantitative results of this configuration, and a graphical depiction of the evolution of session cost thereof.

This configuration is like Shattered Document, except the initial state of the document is a a random connection of each noogramicle to ten others.

We observe that the rate of evolution is small, but the absolute values are very good. The starting session size is already better that the average of 4.46 for the original structure. The final size is similar to Shattered Document.

This result is interesting because it indicates that a legacy, authored structure might be irrelevant for adaptation.

We observe that the session size results differ slightly from the session cost results. We currently have no definitive explanation for this.

4.4. Markov Chains

See figure 7 for the quantitative results of this configuration, and a graphical depiction of the evolution of session cost thereof.

This configuration is like Shattered Document, but using a standard technique for learning the user patterns, namely Markov Chains (first order) [6], instead of our nominal technique of memorizing all user travels individually.

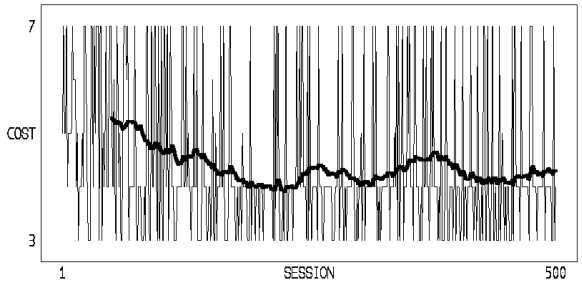
We observe that the results are positive, but not as good as Shattered Document (or Shattered Random Document).

5. Conclusions

The results show that adaptation improves utilization. This was the main result of our work.

Figure 7. Results of configuration Markov Chains

Success_Rate	=	0.84
$Size_0$	=	5.84
$Size_1$	=	4.22
Size_gain	=	1.38
$Cost_0$	=	5.84
$Cost_1$	=	4.22
Cost_gain	=	1.38



The good values for a random document were a surprise, and warrant more investigation. The simulator was also a by-product of note of this work. It allows to experiment with *many* different configurations, as opposed to experimentation with human users, which is much more costly. Albeit based on sensible and established rules of user behaviour, the simulator still lacks explicit validation as a reliable surrogate of human users. These are possible avenues for future work.

Finally, let us highlight the concept of *renoding* as a contribution of this work to adaptive hypertext thinking.

References

[1] J. Ahn, P. Brusilovsky, and R. Farzan. Investigating users' needs and behaviors for social search. In *Workshop on New Technologies for Personalized Information Access (PIA 2005). At 10th International Conference on User Modeling.*, 2005.

[2] M. A. Alves. *Adaptive Hypertext. The shattered document approach*. PhD thesis, Faculdade de Ciências da Universidade do Porto, forthcoming 2013.

[3] M. A. Alves and A. Jorge. Minibrain : a generic model of spreading activation in computers, and example specialisations. In *Relational Machine Learning : ECML-2005 Workshop on sub-symbolic paradigms for learning in structured domains*, page 10 p., 2005.

[4] *Ada Reference Manual, ISO/IEC 8652:200y(E) Ed. 3. — Ada Reference Manual ISO/IEC 8652:1995(E) with Technical Corrigendum 1 and Amendment 1 (Draft 16) : Language and Standard Libraries — Copyright (C) 1992,1993,1994,1995 Intermetrics, Inc. Copyright (C) 2000 The MITRE Corporation, Inc. Copyright (C) 2004, 2005 AXE Consultants Copyright (C) 2004, 2005 Ada-Europe*.

[5] J. Bollen. *A Cognitive Model of Adaptive Web Design and Navigation. A Shared Knowledge Perspective*. PhD thesis, Faculteit Psychologie en Opvoedkunde, Vrije Universiteit Brussel, Belgium, June 2001.

[6] J. Borges and M. Levene. Mining users' web navigation patterns and predicting their next step. In C. Gal, P. Kantor, and B. Shapira, editors, *Security Informatics and Terrorism: Patrolling the Web*, volume 15 of *Science for Peace and Security Series: Information and Communication Security*, pages 45–55, 2008.

[7] C. Mancini. Towards cinematic hypertext: A theoretical and empirical investigation. Technical Report kmi-04-6, Knowledge Media Institute, The Open University, March 2004.

[8] G. A. Miller. The magical number seven, plus or minus two : Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

[9] T. Nelson. A file structure for the complex, the changing and the indeterminate. In *Proceedings of the ACM National Conference*, 1965.

[10] C. Olston and E. H. Chi. Scenttrails: Integrating browsing and searching on the web. *ACM Transactions on Computer-Human Interaction*, 2003.

[11] J. Raskin. *The Humane Interface : new directions for designing interactive systems*. Addison-Wesley, 2000.