Chapter 4

A Formal Approach to Consistency and Reuse of Links in World-Wide Web Applications

Abstract This chapter addresses the problem of evaluating hyperdocuments. It presents a framework for perusing hyperdocuments regarding their structure, aiming to improve the quality of their link organisation. It describes a tool which checks for consistency and reuse of link components, presents some empirical results of its use and proposes a formalism which provides a systematic method of approaching consistency and reuse through link recovery and examination.

1. Introduction

It is widely known that hypermedia systems, which are strongly based on human computer interaction, have been accepted as an innovative framework which plays an important role in current computing environments.

Hypermedia design is a human activity that is complex mainly due to the large amount of information to be processed. The lack of guidelines on what factors should be considered when creating a hyperdocument makes the task even harder. Since there is no consensus on what methodology to use in order to produce a "good" hyperdocument, it is very common to find a poorly written hyperdocument lacking a well organised structure. It is interesting to note, however, that it is difficult to define exactly what "good" means and, consequently, it is very hard to measure "how good" a document is. Due to the fact that hypermedia engineering is currently emerging and therefore still lacks formalisation, hyperdocument authorship is yet to be well defined and is generally not approached in a very systematic way.

In literature there are a large number of different metrics, many of them subjective, that are used as criteria for evaluating a hyperdocument (Brown 1990, Hatzimanikatis et al. 1995). An adequate design-oriented subset of them proposed in (Garzotto et al. 1995) consists of: richness, ease, consistency, self-evidence, predictability, readability and reuse.

This chapter focuses on the formal aspects of two of these metrics, namely, consistency and reuse. The notions of reuse and consistency as stated in (Garzotto et al. 1995) are: "reuse considers the use of objects and operations in different contexts and for different purposes" and "consistency measures application regularity". It is well known, from the field of software reuse in Software Engineering, that appropriate software reuse leads to improvements during the software development phase. In the context of hyperdocuments authoring, according to (Garzotto et al. 1995), an appropriate reuse of objects promotes consistency, predictability and richness.

This chapter offers a tool for the automated evaluation of a web site based on an analysis of the links of the site, and formalises the various cases of consistency and reuse of links that are detected. It is organised as follows: Section 2 highlights the importance of analysing a hyperdocument taking into account its structural and content levels. In Section 3 we propose a systematic approach for examining a hyperdocument by analysing its link structure, and in Section 4 this approach is formalised. The functional structure of a tool named LiOS (**Li**nk-**O**riented **S**ystem) is briefly presented in Section 5, which describes its three main modules. The results from some experiments with LiOS are discussed in Section 6 and our conclusions are presented in Section 7.

2. Perusing the Anchoring Layer

In order to define which measurements are important in the evaluation of a hyperdocument, it is necessary first to define, among its main characteristics, those that can be measured and among them, those which are relevant to be measured, and then establish an appropriate range of values for them.

Following the Dexter reference model for hypertext systems proposed in (Halasz, and Schwartz 1990), hyperdocuments will be considered in this work at two distinct levels of granularity: at the content level and at the structural level. The Dexter model also defines a third level (*presentation layer* in hypertext systems), which is orientated to the resulting forms of display at runtime. Although this chapter is mainly concerned with the analysis of the hyperdocument structure, we believe that by providing ways of improving its structure, we are indirectly contributing to the improvement of the presentation layer, which is the ultimate issue of this book. At the content level (*within the component layer*), the informational contents of the nodes and the existing interrelations within each single node, such as paragraphs and titles, are focused. At the structural level (*the storage layer*) the focus is directed at the interrelations between nodes and links.

A hyperdocument can be independently approached at any of these levels of granularity. In (Salton et al. 1994) the writers focus on the content level of a hyperdocument in order to propose a strategy which automatically defines its links. The work described in (Botafogo and Shneiderman 1992) and in (Rivlin et al. 1994) deals with its structural level, analysing the connectivity of the graph which describes the hyperdocument and proposing algorithms for the identification of possible aggregations, hierarchies and metrics.

In order to approach structural consistency and reuse in a systematic way, we had to take into account both levels of granularity of a hyperdocument. This was due to the fact that the identification of link reuse is strongly supported by anchors, which can be considered part of an interface layer (named *"Anchoring"* in the Dexter model), that exists between the structural level and the content level. By using the Anchoring mechanism, the user interacts with the hyperdocument by means of the hypertext system. It should be pointed out that Anchoring is a mechanism specific to the hypertext system and should also be evaluated with relation to its contribution to hyperdocument quality. The analysis of the Anchoring layer can subsidise a subsequent improvement of the human-computer interaction by means of the improvement of the hyperdocument itself.

3. The Link Oriented Approach for Checking Consistency and Reuse

It is well known in literature that a hyperdocument structure can be seen as a graph (Botafogo and Shneiderman 1992, Stotts et al. 1992). It is common to find research work using this formal view of a hyperdocument structure in order to facilitate the analysis of some of its major characteristics. We have adopted the formal graph view when developing the LiOS tool, since LES[1], which is used for describing the hyperdocument structure, can be seen as a graph description language.

It should be noted, however, that in order to be able to check for consistency and reuse, we had to approach the hyperdocument structure under a *relational view*, as well. Only by adopting a relational view could the *virtual structures* we identified, namely *aggregation* and *degeneration*, be detected. It is worth noting that, while trying to identify the possible virtual structures, three issues were considered:

1. existing techniques for node naming - the Universal Resource Locator (URL) of a document used in WWW is already an established and accepted technique for this;

2. the fact that there is no established technique for naming links. This gives rise to a certain degree of freedom to the author of the hyperdocument that might possibly make the hyperdocument more difficult to manage;

3. the fact that the larger the volume of information in a hyperdocument is, the higher the probability for reuse of nodes and links to occur.

In order to make a systematic and exhaustive search for all possible virtual structures, a tree representing the possible reuse cases was created, as shown in Figure 1. The reuse cases were defined based on unidirectional links. The definitions were supported by the results obtained by analysing the reuse of the components of unidirectional links, namely: **source-node, destination-node** and **source-anchors**[2]. Each of these components was approached through reuse, with respect to other links, in the context of a hyperdocument. By associating a binary valued reuse attribute (0:no and 1:yes) to each of the three components, the possible reuse cases of link components added up to 8 (2^3), ranging from what was identified as case 0 up to case 7.

The tree structure in Figure 1 represents the eight possible cases which resulted from combining the components of the links, starting with links which have a different source-node, different source-anchor and different destination-node (case 000_2,

[1] LES (Statechart Language Specification) is a language which allows the textual description of the structural components of a hyperdocument, according to the Statechart syntax and semantics (Harel 1987).

[2] It is a common practice that anchors in source-nodes and anchors in destination-nodes are treated differently. An anchor is necessary in the source-node since it is the starting point of a link to another node (generally). At the arrival node, in general, an anchor is an entry point to that node and it does not need to be presented to the reader.

labelled 0) up to those which have the same source-node, same source-anchor and same destination-node (case 111_2, labelled 7).

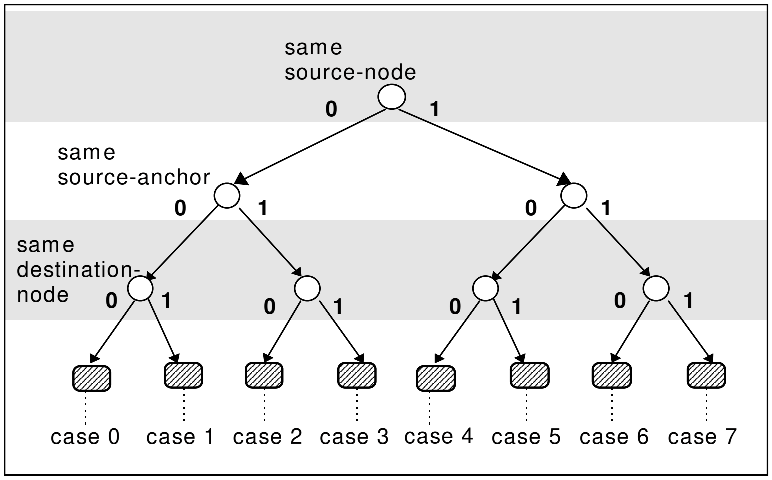


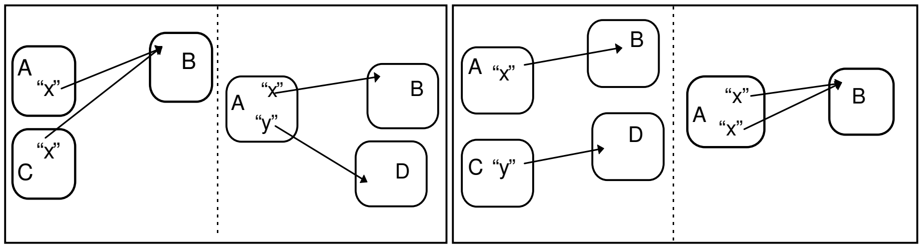
Figure 1 - Tree representing the reuse cases of link components

Using the formal graph structure of a generic hyperdocument, as defined in (Parunak 1991), as well as the semantic attached to link reuse, a classification of the possible reuse cases of link components was carried out. Three main reuse situations were identified and the corresponding links classified either as **general**, **unusual** or **contextual**. The empirical results presented in Section 6 reinforce the fact that the proposed taxonomy of links can be adopted as a sound metric for evaluating hyperdocuments.

A semantic view of the eight identified cases, grouped by their classification, is presented in Figure 2 (a), (b) and (c), where the rounded rectangles represent nodes named A, B, C and D, the arrows represent the links and labels "x" and "y" correspond to the value of two different source-anchors.

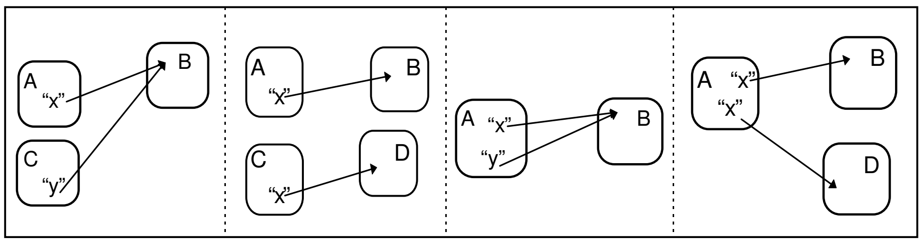
Cases 3 and 4 are considered **general links**; case 3 can be interpreted as an abstraction of index pages (more than one link with the same source-anchor arriving at the same destination-node) and case 4 as an abstraction of table-of-contents page (more than one different link leaving the same source-node).

Case 0 and case 7 are categorised as **unusual links** in the context of link reuse since they represent the unusual situations of no reuse and of total reuse, respectively. The identification of a link reuse belonging to case 0 or case 7 suggests the need for a deeper analysis, since each case corresponds to an "extreme" case.



case 3 $(011)_2$ case 4 $(100)_2$ case 0 $(000)_2$ case 7 $(111)_2$

(a) General links (case 3 and case 4) **(b) Unusual links** (case 0 and case 7)



case 1 $(001)_2$ case 2 $(010)_2$ case 5 $(101)_2$ case 6 $(110)_2$

(c) Contextual links (cases 1, 2, 5 and 6)

Figure 2 - Classification of link cases based on their component reuse

Finally, cases 1, 2, 5 and 6 are classified as **contextual links**. Apparently, the situations described by these cases could be considered abnormal, but before doing so, a closer examination of the node itself (its contents and the region which surrounds the anchor) should be carried out in order to ascertain its abnormality.
Note that, in odd-numbered cases, we are considering links which, besides having the same destination-nodes, arrive at the same entry-point.
To get a more generic view of the proposed taxonomy, a study of the possible abstractions that exist or could be created in each of the eight identified cases was conducted, aimed at:

- generalising the link classification procedure, and
- providing alternative ways of detecting virtual structures that can be implicitly defined during the authoring process.

Based on a careful and systematic study of all possible abstractions, two generic procedures for creating an abstraction were identified:

1. ***aggregation*** (identified by “***g***”): when the derived cases are created by grouping destination-nodes;
2. ***degeneration*** (identified by “***d***”): when a link which leaves and arrives at the same node is created.

It must be noted, however, that aggregation only deals with destination-nodes; this is due to the fact that grouping of source-nodes has already been considered previously (cases 4, 5, 6 and 7). Both procedures are illustrated in Figure 3, where case 5 is derived from case 1 by aggregation of source-nodes and case 5d derives from case 5 by degeneration.

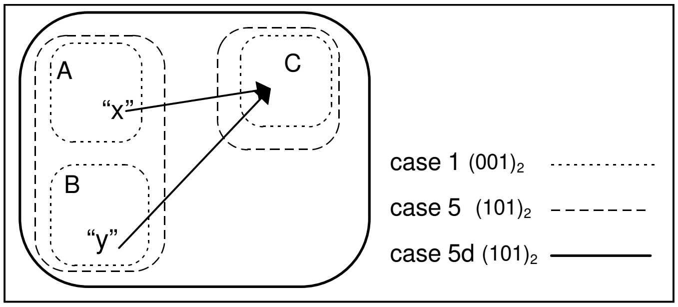


Figure 3 - Case 1 leads to case 5 by aggregation and case 5 leads to case 5d by degeneration

To exemplify our systematic approach let us take as an example case 0, which identifies situations where links have different source-nodes, different destination-nodes and different source-anchors. A complete view of the whole derivation process is presented in Figure 4.

As stated earlier, case 0 has no reuse of any link component. By exhaustively deriving all possible cases (using aggregation, degeneration or a combination of both), case 4 has been identified as one among many possible derivations from case 0. The links described in case 0 and case 4 are characterised by having different source-anchors and different destination-nodes. What distinguishes case 0 from case 4 is the fact that the source-anchors are placed in different nodes (case 0) and in the same node (case 4). Case 4 represents grouping of the source-nodes displayed in case 0, namely, A and C, in just one node, A.

Two other aggregations were carried out from case 0 and case 4 by grouping the destination-nodes: one identified as case 0g and the other as case 4g, respectively. The degeneration procedure applied to case 0, case 4 and case 0g gave rise to case 0d, case 4d and case 0gd respectively. One of the two other cases not yet mentioned, which is in the last level of the tree pictured in Figure 4, can be obtained by applying degeneration to case 4d or equivalently, by applying degeneration to case 4g. That is why it can be identified as both: case 4gd or case 4dd. The last one, case 0gd is derived by applying aggregation to case 0d.

Finally, after going through the abstraction process for each of the eight cases, cases 0, 1, 2 and 3 were chosen as **primitive cases**, because they give rise to others by aggregation or degeneration and they are not derived from any other case by means of the same operations.

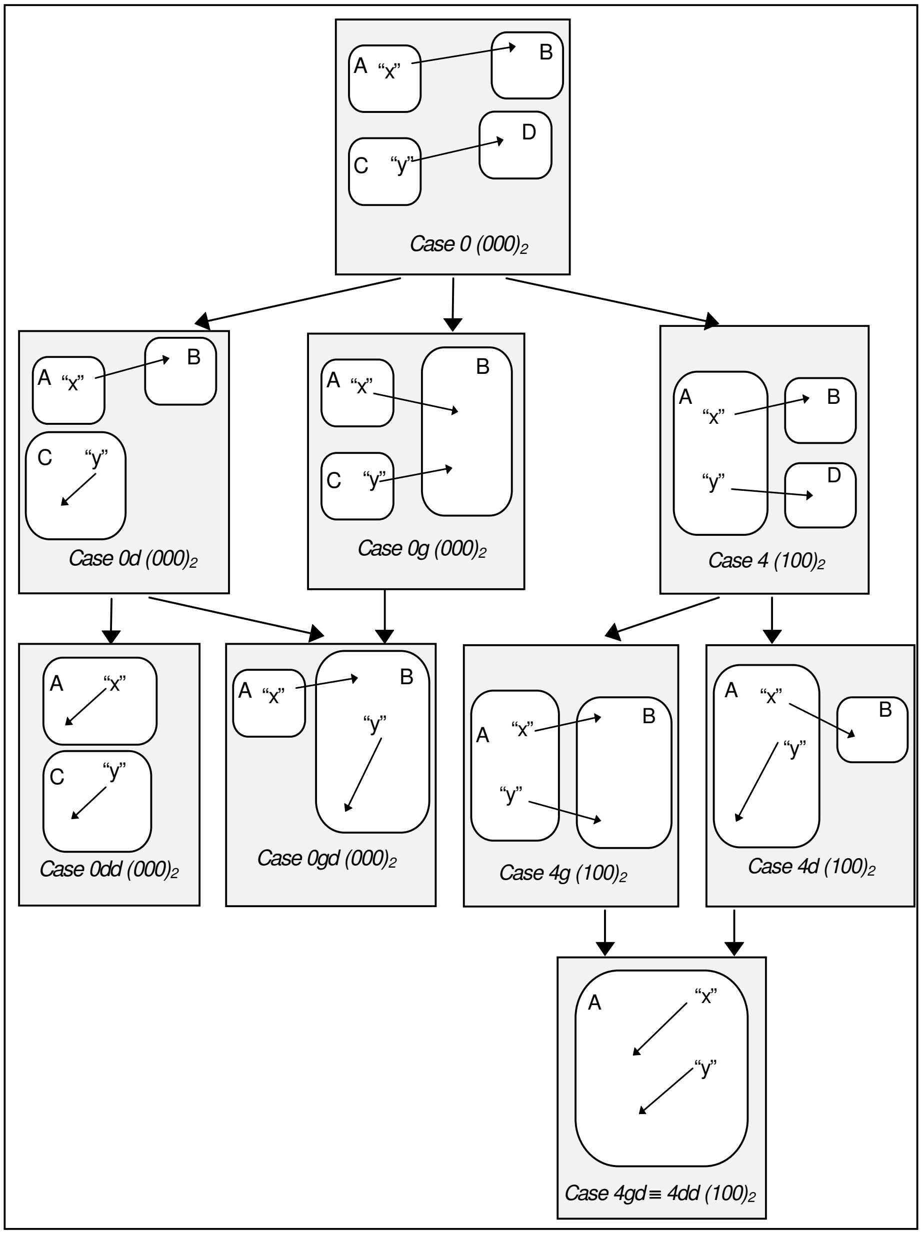


Figure 4 - Cases derived from case 0

Based on the primitive cases, following the systematic approach exemplified earlier, all the possible abstractions of those cases were derived by means of degeneration, aggregation or a combination of both, as shown in Table I. It can be noted that, by allowing virtual structures (aggregation, degeneration or both), 20 new cases can occur.

Table I - Primitive cases and their possible derivations through aggregation, degeneration or both

primitive cases	derived cases by *aggregation*	derived cases by *degeneration*	derived cases by both
case 0	0g, 4, 4g	0d, 4d, 0dd	0gd, 4gd
case 1	5	5d	—
case 2	2g, 6, 6g	2d, 6d, 2dd	2gd, 6gd
case 3	7	7d	—

The next section presents a formalization of the systematic approach for link analysis just presented, where each of the eight identified cases of link component reuse is formally described.

4. Recovering Links: A Formal Approach

In this section we begin by introducing the formal definition of a hyperdocument used as a basis for establishing the formalisation of link reuse cases. According to (Parunak 1991), a graph-based hyperbase is an ordered triplet $H = <N, A, L>$ of nodes, anchors and links respectively, where:

- $N = \{N_1, ...N_m\}$ is a set of information nodes
- $A = \{A_{11}, ...A_{mp}\}$ is a set of anchors, where each node N_i has one or more anchors $\{A_{i1}, ..., A_{iq}\}$
- $L = \{<A_{ij}, A_{kl}>\}$ is a set of links, i.e., a set of ordered pairs of anchors

In order to use this definition for the purpose of link reuse formalisation, the following observations should be taken into account:

- N can be unitary. This happens when the hyperdocument has only one node. Obviously the set $N = \{N_1, ...N_m\}$ should be finite, where m represents the number of nodes belonging to N.
- A link also can be defined between two anchors in one same node, i.e., some $<A_{ij}, A_{il}>$, where A_{ij} and A_{il} are anchors placed on node N_i.
- The set of anchors of node N_i will be noticed by A_i. If node N_i has p anchors, i.e., the set of its anchors is $\{A_{i1},..., A_{ip}\}$, the number of elements of A_i is p, i.e., $|A_i|=p$
- $L \subseteq A \times A$ and does not exist $<A_{ij},A_{ij}>$, for $1\leq i \leq m$ and $|L| \geq 1$
- Let $<A_{ij},A_{kl}> \in L$. The anchor A_{ij} is named source-anchor and A_{kl} destination-anchor

To identify the links belonging to each link reuse case, we extended Parunak's definition by introducing two new functions:

1. *source-anchor*: $L \rightarrow A$ such that *source-anchor*($<A_{ij}, \_>$) = A_{ij}, a function which extracts the source-anchor from a link
2. *name*: $A \rightarrow$ <*string*> | <*icon*>, which maps an anchor to its value (or its icon) presented to the user during navigation

Next, a set-based description of each of the eight identified cases (depicted in Figure 1 and in Figure 2), is presented. We have chosen to introduce some redundancy in the notation, aiming to make it clearer and more uniform. That is why, for example, we write $<N_r, N_t>$, $r = t$, instead of $<N_r, N_r>$.
Thus, for $1 \le i, j, r, t \le m$, and $1 \le k \le |A_i|$, $1 \le l \le |A_j|$, $1 \le s \le |A_r|$, $1 \le x \le |A_t|$, the set-based descriptions are:

<u>Case 0</u> $(000)_2$ -
(0a) $\{< <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) \neq \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$i \neq j, r \neq t \}$
(0b) $\{< <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) \neq \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$i \neq j, r = t, s \neq x \}$

<u>Case 1</u> $(001)_2$ -
(1) $\{ < <N_i,N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) \neq \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$i \neq j, r = t, s = x\}$

<u>Case 2</u> $(010)_2$ -
(2a) $\{ < <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) = \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$i \neq j, r \neq t \}$
(2b) $\{< <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) = \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$i \neq j, r = t, s \neq x \}$

<u>Case 3</u> $(011)_2$ -
(3) $\{ < < <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\mathit{name}(\mathit{source\text{-}anchor}(<A_{ik},A_{rs}>)) = \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl},A_{tx}>)),$
$i \neq j, r = t, s = x\}$

<u>Case 4</u> $(100)_2$ -
(4a) $\{< <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) \neq \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$i = j, k \neq l, r \neq t \}$
(4b) $\{< <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) \neq \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$i = j, k \neq l, r = t, s \neq x \}$

<u>Case 5</u> $(101)_2$ -

(5) $\{ < <N_i,N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\quad \mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) \neq \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$\quad i = j,\ k \neq l,\ r = t,\ s = x\}$

<u>Case 6</u> $(110)_2$ -

(6a) $\{ < <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\quad \mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) = \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$\quad i = j,\ k \neq l,\ r \neq t\ \}$

(6b) $\{< <N_i, N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\quad \mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) = \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$\quad i = j,\ k \neq l,\ r = t,\ s \neq x\ \}$

<u>Case 7</u> $(111)_2$ -

(7) $\{ < <N_i,N_j>, <A_{ik}, A_{jl}>, <N_r, N_t> > \mid$
$\quad \mathit{name}(\mathit{source\text{-}anchor}(<A_{ik}, A_{rs}>)) = \mathit{name}(\mathit{source\text{-}anchor}(<A_{jl}, A_{tx}>)),$
$\quad i = j,\ k \neq l,\ r = t,\ s = x\}$

It can be observed that the existence of sets a) and b) for the even-numbered cases (which do not have same destination) represent the occurrence of destination-anchors placed in a) different destination-nodes, and b) in the same destination-node, but arriving at different entry-points.

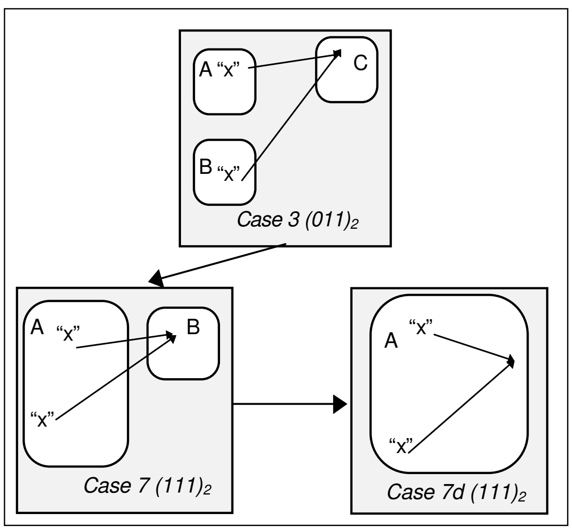


Figure 5 - Case 7 and case 7d derived from case 3

The formal approach for describing each of the eight identified cases also covers the formal description of any of the 16 possible derivations obtained from case 0 up to case 7. For example, case 7d illustrated in Figure 5, is a degeneration of case 7, because the source-node and the destination-node of case 7 have become a single node which is both the source-node and the destination-node of case 7d. It is important to note that case 7d can also be described using the same set-based description that describes case 7; of course, the restriction of having the same node

for source and destination-node is offset by requesting that i = r. This can be seen as a specialisation of the formal description given for case 7.
Table II, below, gives the link reuse cases covered by it for each set-based description presented.

Table II - Set-based descriptions and the corresponding link reuse cases they cover

description	*covers*
(0a)	0, 0d, 0dd
(0b)	0g, 0gd
(1)	1
(2a)	2, 2d, 2dd
(2b)	2g, 2gd
(3)	3

description	*covers*
(4a)	4, 4d, 4dd
(4b)	4g, 4gd
(5)	5, 5d
(6a)	6, 6d, 6dd
(6b)	6g, 6gd
(7)	7, 7d

5. Analysing Link Reuse Cases: the LiOS Tool

To implement the consistency and reuse criteria for analysing and evaluating hyperdocuments, we have chosen a *link-oriented approach*, which is based on the structure of the hyperdocument.
To peruse both consistency and reuse of link components, an automatic tool named LiOS (Fortes 1996) was developed, based on the identification and recovery of the following objects: link, source-node, source-anchor and destination-node. These objects are diagrammatically shown in Figure 6, using the notation proposed in (Coleman et al. 1994).

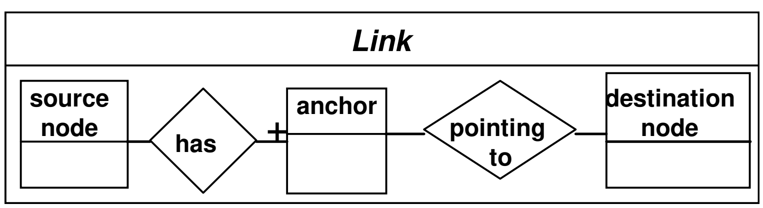


Figure 6 - The components of an object "link"

The modular functional structure of the tool was defined, based on this object model, as shown in Figure 7. The structure is basically composed of three main modules: the *HD Representation Module*, the *Converter Module* and the *Presentation Module*. A brief description of each of these modules is shown below.
The *HD Representation Module* scans the hyperdocument following its local links and generates an ASCII file, containing the structure of the hyperdocument described in LES (Fortes and Masiero 1991). The structure itself can be seen as a set of links, each of them described as a Statechart transition. To describe the links in LES, we used the convention that an anchor, presented to the reader in the source node, corresponds to an event that might possibly be triggered by him/her, i.e., it corresponds to a link.

Since the scanning process does not follow links through the HTTP protocol, it produces a local hyperdocument structure representation containing only the links belonging to the author's production. An important characteristic of the functionality of the *HD Representation Module* is related to the URL address decoding process, aiming at standardisation, in order to identify the real reuse of link components.

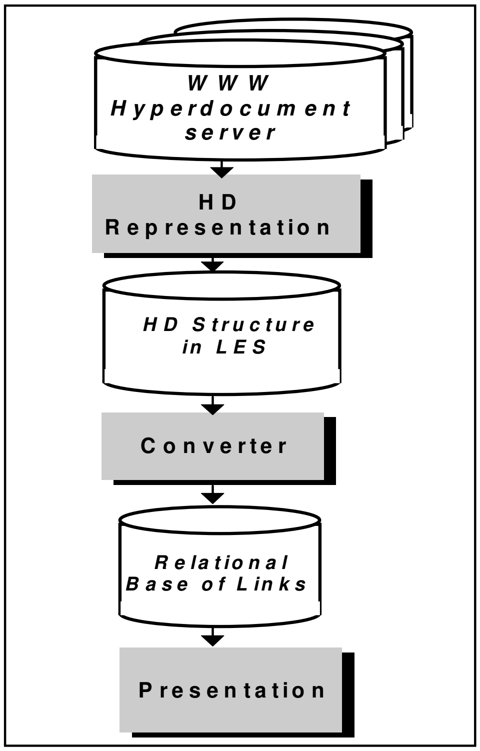


Figure 7 - Functional structure of LiOS, a link-oriented tool

The *Converter Module* is responsible for converting the ASCII file, generated by the *HD Representation Module*, into a relational data base, where links are represented as t-uple having the form:

(<u>id-link</u>, source-anchor, source-node, destination-node, destination-anchor)

where:

id-link = link-identifier (primary-key);
source-anchor = event;
source-node = source;
destination-node.destination-anchor = destination.

The *Presentation Module* is responsible for presenting the user with a list of all the existing links in the hyperdocument, so that they can be selected for further examination. The module focuses on the links and then on the nodes they connect. This module's main window shows a list of all the source-anchor names used in the hyperdocument and makes them available for selection. Selecting any of these anchor names gives the user access to two new lists: the first, containing all the source-nodes which have, as their starting point, the selected anchor and the second, containing all the destination-nodes which can be reached from that same anchor. Figure 8 shows a screen dump of the *Presentation Module*, where "Eventos", "Origem" and "Destino" are the Portuguese words for events (meaning source-anchors), source (source-node) and destination (destination-node).

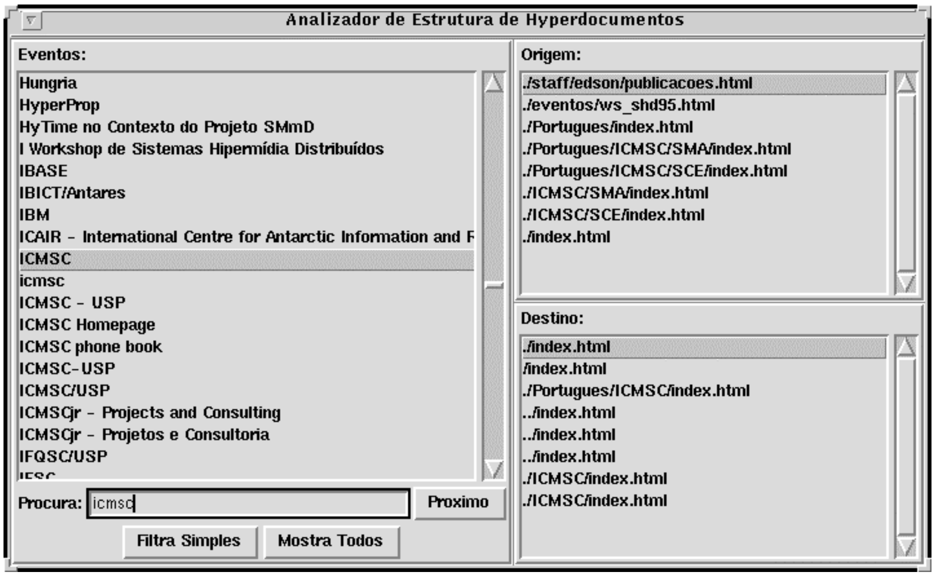


Figure 8 - Window presenting the links of a WWW hyperdocument, showing the source and destination nodes which are connected by them

To understand how the *Presentation Module* works let us imagine the situation pictured in Figure 9, where the link named "here" links document Doc1 to Doc11 and Doc12 and another link, also named "here", links document Doc2 to Doc21. Clicking on link "here" would result in documents Doc1, Doc1 and Doc2 being shown as source-nodes and Doc11, Doc12 and Doc21 being shown as destination-nodes.

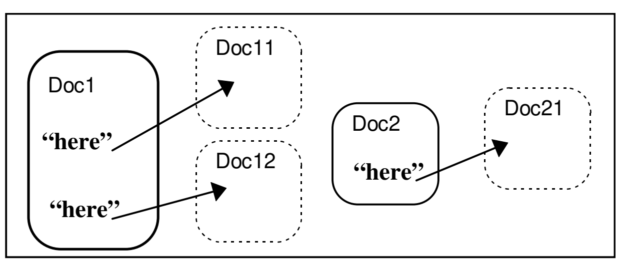


Figure 9 - An example of a possible situation dealt with by the *Presentation Module*

6. Experimenting with LiOS — Empirical Results

It is worth reminding the reader again that this chapter focuses mainly on a formal model of hypermedia structure, and addresses link reuse cases rather than formal methods for the design of interactive systems, which is the main issue of this book. Nevertheless, we believe that, since both fields are closely interrelated, this chapter offers a useful contribution toward evaluating a hyperdocument insofar as link reuse is concerned, and it can therefore help improve the quality of a hyperdocument, with obvious benefits to computer-human interaction. Indeed, the exercise of blending these two fields has been gaining increased interest on the part of the HCI community as can be seen, for instance, in (Yamada 1995).

The systematic link reuse approach to hyperdocument analysis, its formalisation and its implementation were carried out already including WWW applications in mind. The project this chapter describes was proposed and designed for the analysis of WWW hyperdocument applications. It was assumed, right from the inception of this study, that a hyperdocument in its HTML description would be the main focus of the project, considering we were mainly concerned with the improvement of its structure. Thus, as far as the Web Browser Case Study is concerned, we do not have much to add to what we have pointed out earlier, that is: by improving the hyperdocument structure, we indirectly improve its visualisation and other features that a Browser can make available.

In order to evaluate its contribution, the LiOS tool was experimentally used in two different sites. The hyperdocument of one site (Instituto de Ciências Matemáticas de São Carlos - Universidade de São Paulo, site ***http://www.icmsc.sc.usp.br***) is identified in this work as ICMSC-site and the hyperdocument of the other site (Instituto de Física de São Carlos - Universidade de São Paulo, site ***http://www.ifsc.sc.usp.br***) as IFSC-site. We would like to point out that LiOS is currently operating with a specified set of hyperdocuments, i.e., it uses only "local" links of a certain site. In order to deal with links from more than one site, the tool must be adjusted to accept the other site's address as a parameter.

Both websites used in our experiment can be considered "good" hyperdocuments for checking the link reuse approach, mainly due to the fact that:

- they have several webmasters; pages can be created and maintained by the administrative staff, lecturers, researchers and students. Since some of the webmasters are students, responsibility for the management of the whole hyperdocument frequently changes over from one person to another;
- the volume of nodes and links in these webs are considerable, reaching the thousands;
- the ICMSC-site and the IFSC-site were created in mid-1994 and are among the first produced in Brazil;
- the information available in both hyperdocuments is quite heterogeneous and, despite the fact that some of the basic information is kept permanently, they have been constantly altered by the inclusion/removal of nodes and links (temporary lectures, exercises, manuals, etc.).

The main goal in conducting a few empirical experiments was to gather data to support the research and analysis of reuse cases in order to get feedback concerning the applicability and usefulness of the approach implemented by LiOS. The whole experiment intended, as well, to compare the evolution of reuse cases over a period of time, in an attempt to discover a possible bias in hyperdocument maintenance procedures.

Data was collected on three different occasions (September-95, November-95 and July-96) from the ICMSC-site, and are referred to in this work as ICMSC/Sep-95, ICMSC/Nov-95 and ICMSC/July-96 respectively, and on just one occasion (July-96) from the IFSC-site, which is referred to in this work as IFSC/July-96.

Table III shows the counts obtained from queries about the incidence of links in each of the reuse cases identified previously. The results reveal important information about how links have been used in both hyperdocuments. For a better understanding of the data presented in Table III, some conventions must be explained:

- the value of R represents the total of distinct occurrences of links in each relation resulting from a query in SQL. It corresponds to the value returned by the function COUNT( ), which counts the number of registers in a table. More specifically, the value of R corresponds to the number of types of links which have a certain reuse (the type of reuse is defined by the grouping operator (*group by*) specified in the query);
- the number of effective links given by the value of L corresponds to the value returned by the function SUM( ), which adds up the number of incidences for each type of grouped link;
- to count the incidence of links belonging to each case we took into consideration the “subset” relation. By using this relation, one particular link can be seen as belonging to more than one case.

It should be noted that the percentages of all cases added up will not amount to 100%; this is because links are not considered exclusively as belonging to just one case. A certain node can play the role of source node in different types of links. Only the set of incidences of links classified as case 7 does not contain any of the other sets of links (associated with other cases). All the other cases, from 0 to 6, can either contain or be contained in another case.

Table III - Counts from queries about the incidence of links of each one of the reuse cases of link components — experiments with ICMSC-USP and IFSC-USP WWW servers

case	ICMSC-site September/95	ICMSC-site November/95	ICMSC-site July/96	IFSC-site July/96
	2,334 links 1,406 nodes	2,433 links 1,419 nodes	8,066 links 1,201 nodes	3,255 links 546 nodes
0	R=19 L=19 (0.81%)	R=18 L=18 (0.74%)	R=26 L=26 (0.32%)	R=46 L=46 (1.41%)
1	R=127 L=413 (17.69%)	R=118 L=361 (14.77%)	R=426 L=1,583 (19.62%)	R=135 L=503 (15.45%)
2	R=89 L=281 (12.03%)	R=83 L=196 (8.02%)	R=356 L=1,198 (14.85%)	R=101 L=362 (11.12%)
3	R=227 L=649 (27.80%)	R=261 L=712 (29.14%)	R=1,076 L=3,541 (43.9%)	R=348 L=1,200 (36.86%)
4	R=155 L=1,674 (71,72%)	R=169 L=1,735 (71.01%)	R=801 L=7,117 (88.23%)	R=339 L=2,678 (82.27%)
5	R=86 L=216 (9.25%)	R=82 L=209 (8.55%)	R=165 L=358 (4.43%)	R=93 L=217 (6.67%)
6	R=29 L=182 (7.79%)	R=16 L=150 (6.13%)	R=55 L=257 (3.18%)	R=36 L=76 (2.33%)
7	R=34 L=140 (5.99%)	R=36 L=145 (5.93%)	R=83 L=244 (3.02%)	R=26 L=54 (1.66%)

Observing the results in Figure 10, obtained from the data ICMSC/Sep-95, ICMSC/Nov-95 and ICMSC/July-96, we can see that the volume of links increased over time. Although it is rather premature to state that this tendency will always occur, informally it can be justified by noting that, after a while, authors gradually feel more confident and begin improving their documents by creating new links. A closer look at Figure 10 allows us to state that:

- The links classified as general links are the most common kind found in a hyperdocument structure;
- The links classified as unusual links (case 0 and case 7) are those which would rarely be found in a hyperdocument structure. Nevertheless, the number of such links was greater than expected; this may be due to the fact that many nodes,

mainly in the ICMSC-USP-WWW, resemble very long textual documents full of cross references;

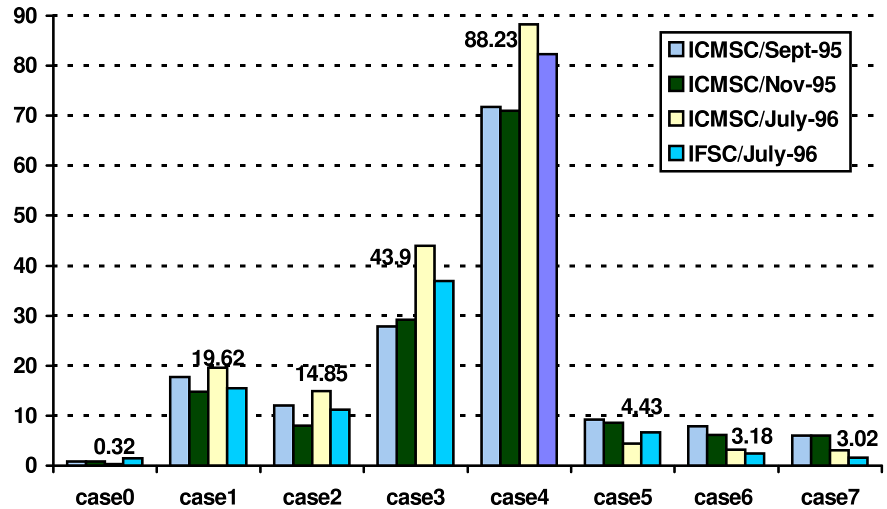


Figure 10 - Percentage of links per case

- The contextual links (case 1, 2, 5 and 6) are the least common cases expected to be found among the links. Despite this fact, however, two opposing tendencies can be observed in regard to their use:
  - Where "modularity" is concerned, the reuse of objects (nodes or pages) is a very important and emphasised practice. In the context of modularity, therefore, an increase of contextual links suggests a "possibly good" hyperdocument when used in an appropriate way.
  - Since contextual links somehow promote ambiguity and/or inconsistency, there is a tendency for this kind of link to be avoided and consequently, for its frequency to decrease. Thus, the decrease in the number of contextual links may be evidence that the team of authors has been concerned with the quality of the hyperdocument.

What can be stated from those two opposing views is that an increase or decrease in the number of contextual links is inconclusive and that additional information must be provided in order to be able to make a conclusive statement about the quality of the hyperdocument, as far as link analysis is concerned.

It is worth mentioning that the empirical results obtained, i.e., the percentage of link cases, were presented to the webmasters in order to get some feedback about the relevance of those numbers. Their comments lead us to conclude that some problems at a website can be detected with the tool. According to them, for example:

"— Many **case 0** links were detected due to the fact that a hypertext in HTML has only unidirectional links. As pages are presented by systems which have backtrack buttons, like Mosaic and Netscape, many of them do not have any backtrack link and therefore they have no reuse. Generally speaking, however, those occurrences are avoided."

"— Many **case 1** links could be redefined for the sake of uniformity and could, consequently, be "transformed" into case 3 links. Nevertheless, few of those links were created deliberately to reuse the contents of destination pages."

"— The reuse of link names (source-anchors) as seen in **case 2** is very practical to indicate a sequential direction; this is typical when the author wants the reader to follow a certain pre-defined "path" (e.g. classroom notes). On the other hand, when the hyperdocument is read looking for a specific piece of information, this kind of reuse can cause disorientation."

"— The more **case 3** links there are the better, because this implies uniformity of terms."

"— The high occurrence of **case 4** links is due to index-pages. It would be very interesting to perform a study concerning the rate of distribuition of links in relation to the number of distinct records."

"— The use of different links on the same pages should be avoided. Many of the existing occurrences of cases such as **case 5** links come from newcomers who use the WWW to exchange information."

"— It seems that **case 6** links are a consequence of the direct transcription of texts into hyperdocuments; the hyperdocument often inherits the reuse of words from the original text (which employs reuse in order to prevent word overload)."

"**— Case 7** links are generally avoided when identical links are close together on the page in order to prevent a "polluted" document. Case 7 links usually happen in hyperdocuments which originate from printed manuals; the long pages in the hyperdocument reflect the pages of the manual printed on paper."

7. Conclusions

This paper describes a link-oriented approach which has been used for webmasters to peruse the consistency of components in links. The formalisation of this approach is also presented. We have also shown how a classification of reuse of link components was systematically created. The main conclusion we have reached so far is that the identification of virtual structures and their formalisation has provided the authorship process with several alternatives for reorganising a hyperdocument.

The main goal of the work described in this paper was to provide means to help a hyperdocument writer to produce and maintain a consistent document. We believe that, by identifying the virtual structures, the author will have alternative views for reorganising the hyperdocument so as to avoid ambiguity and improve navigation. This would contribute to the improved use of the available hypermedia application structures. Use of the framework enables the author to evaluate the organisation, style and uniformity of the hyperdocument he/she is creating.

Also presented herein is the LiOS tool used for analysing the structure of link connections among documents on a web server. LiOS aims to provide help to authors in correcting possible structural problems detected among web pages.

As for future work, we intend to provide mechanisms for identifying the semantic network associated with the hyperdocument structure by expressing its underlying organisation, using virtual structures.

Acknowledgments

This research was partially supported by FAPESP and CNPq (ProTeM III-CC)

References

Botafogo, R.A. and Shneiderman, B. (1992) Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Trans. on Info. Systems*, 10(2):142-180

Brown, P.J. (1990) Assessing the quality of hypertext documents. In: Rizs, A.; Stritz, N. & André, J. (Eds.) *Hypertext: Concepts, Systems and Applications* Cambridge, Great Britain, Cambridge University Press. /Proceedings of *ECHT'90* Versailles, France, November 1990/ pp.1-12

Coleman, D. et al. (1994) *Object-oriented development: the FUSION method*. New Jersey, USA, Prentice Hall, Englewood Cliffs

Fortes, R.P.M. (1996) Analysis and evaluation of hyperdocuments: an approach based on the structural representation. (In Portuguese) *Phd. Thesis*, IFSC-USP, São Carlos - SP, Brazil

Fortes, R.P.M. and Masiero, P.C. (1991) A language for a Statechart specification and its analyser. (In Portuguese) *Technical Report* ME-91/01, ICMSC-USP (http://www.icmsc.sc.usp.br/~renata/WWW/LES.html)

Garzotto, F., Mainetti, L. and Paolini, P. (1995) Hypermedia design, analysis and evaluation issues. *CACM*, 38(8):74-96

Halasz, F. and Schwartz, M. (1990) The Dexter hypertext reference model. In: Moline, J.; Benigni, D.; Baronas, J. (Eds.) Proceedings of the *NIST Hypertext Standardization Workshop*, NIST Special Publication 500-178. Washington: U.S. Government Printing Office, pp 95-133

Harel, D. (1987) Statecharts: on the formal semantics of Statecharts. In: Proceedings of the *2nd IEEE Symposium on Logic in Computer Science*, Ithaca, NY, pp.54-64

Hatzimanikatis, A.E.; Tsalidis, C.T. and Christodoulakis, D. (1995) Measuring the readability and maintainability of hyperdocuments. *Software Maintanance: Research and Practice*, 7:77-90

Parunak, H.Van D. (1991) Don't link me in: set based hypermedia for taxonomic reasoning. In: Proceedings *of Hypertext'91*, San Antonio, TX, pp.233-242

Rivlin, E., Botafogo, R. and Shneiderman, B. (1994) Navigating in hyperspace: designing a structure-based toolbox. *CACM*, 37(2):87-96

Salton, G., Allan, J. and Buckley, C. (1994) Automatic structuring and retrieval of large text files. *CACM*, 37(2):97-108

Stotts, P.D.; Furuta, R. and Ruiz, J.C. (1992) Hyperdocuments as automata: trace-based browsing property verification. In: Proceedings of *ECHT'92*, Milano, December 1992. New York, ACM Press, pp.272-281

Yamada, S., Hong, J. and Sugita, S. (1995) Development and evaluation of hypermedia for museum education: validation of metrics. *ACM Trans. Computer-Human Interaction*, 2(4):284-307